

BridgeVIEW™ et LabVIEW™

Manuel de référence de programmation en G

Filiales francophones

National Instruments Belgium nv	National Instruments Canada–Montréal	National Instruments France	National Instruments Suisse
Leuvensesteenweg 613	1000 Boulevard St. Jean,	Centre d’Affaires Paris-Nord	Sonnenbergstr. 53
B-1930 Zaventem	Suite 316	Immeuble “Le Continental”	CH-5408 Ennetbaden
	Point-Claire, Québec H9R 5P1	BP 217	
		93153 Le Blanc-Mesnil Cedex	

Support internet

E-mail : support@natinst.com
Site FTP : <ftp.natinst.com>
Adresse web : <http://www.natinst.com>

Support Bulletin Board

BBS France : 01 48 65 15 59
BBS Etats-Unis : 512 794 5422

Support téléphonique en français

Belgique	Tél. : 02 757 00 20	Fax : 02 757 03 11	Tél. : 405 120 (Luxembourg)
Canada (Québec)	Tél. : 514 694 8521	Fax : 514 694 4399	
France	Tél. : 01 48 14 24 24	Fax : 01 48 14 24 14	
Suisse	Tél. : 056 200 51 51	Fax : 056 200 51 55	Tél. : 022 980 05 11 (Genève)

Les filiales

Allemagne 089 741 31 30, Australie 03 9879 5166, Autriche 0662 45 79 90 0, Brésil 011 288 3336, Canada (Ontario) 905 785 0085, Corée 02 596 7456, Danemark 45 76 26 00, Espagne 91 640 0085, Finlande 09 725 725 11, Hong Kong 2645 3186, Israël 03 6120092, Italie 02 413091, Japon 03 5472 2970, Mexique 5 520 2635, Norvège 32 84 84 00, Pays-Bas 0348 433466, Royaume-Uni 01635 523545, Singapour 2265886, Suède 08 730 49 70, Taiwan 02 377 1200

National Instruments Corporate Headquarters

11500 North Mopac Expressway Austin, Texas 78759 USA Tél: (+1) 512 794 0100

Informations importantes

Limitations de garantie

Les disquettes ou CD sur lesquels vous recevrez le logiciel de National Instruments sont garantis contre les pannes survenant lors de l'exécution de programmes, qui seraient dues à des défauts matériels ou de fabrication. La période de cette garantie est de 90 jours à partir de la date de livraison, attestée par les reçus ou autres documents. Le cas échéant, National Instruments corrigera ou remplacera la disquette ou le CD qui ne permettrait pas l'exécution normale des programmes, à condition qu'un tel défaut soit stipulé au cours de la période de garantie. National Instruments ne garantit pas que le fonctionnement de ses logiciels ne sera pas interrompu ni se déroulera sans erreur.

Un numéro RMA (Return Material Authorization) d'autorisation de retour de matériel doit être délivré par l'usine et clairement indiqué sur l'emballage du produit afin que celui-ci puisse être accepté pour la réparation sous garantie. National Instruments prendra à sa charge les frais de transport pour renvoyer à son propriétaire les éléments couverts par la garantie.

National Instruments estime avoir fait tout ce qu'il fallait pour que les informations contenues dans ce manuel soient exactes. Ce document a été soigneusement relu pour la précision des informations techniques qu'il contient. Au cas où il resterait malgré tout des erreurs techniques ou des fautes typographiques, National Instruments se réserve le droit d'apporter des modifications aux futures éditions de ce document sans préavis aux détenteurs de cette édition. Le lecteur est prié de prendre contact directement avec National Instruments s'il suspecte des erreurs. National Instruments ne pourra être tenu responsable des problèmes liés à l'utilisation de ce document ou aux informations qu'il contient.

À L'EXCEPTION DE CE QUI EST SPÉCIFIÉ ICI, NATIONAL INSTRUMENTS N'ACCORDE AUCUNE AUTRE GARANTIE, EXPLICITE OU IMPLICITE, ET REJETTE PARTICULIÈREMENT TOUTE GARANTIE LIÉE À L'ACTE DE VENTE ET À L'ADÉQUATION DE SES PRODUITS À UN BESOIN PARTICULIER. LES DROIT DES UTILISATEURS À RECOUVRER LES DOMMAGES CAUSÉS PAR UNE FAUTE OU NÉGLIGENCE DE LA PART DE NATIONAL INSTRUMENTS SERONT LIMITÉS AUX SOMMES VERSÉES PAR L'UTILISATEUR. NATIONAL INSTRUMENTS NE SERA PAS PASSIBLE DE DOMMAGES ET INTÉRÊTS À LA SUITE DE PERTES DE DONNÉES OU DE PROFITS, OU DE DOMMAGES (ACCIDENTELS OU NON) LIÉS À L'UTILISATION DE SES PRODUITS, MÊME S'IL EN AVAIT ÉTÉ PRÉALABLEMENT AVERTI. Cette limitation de la responsabilité de National Instruments s'appliquera quelles que soient la nature et l'origine du préjudice, que ce soit à la suite d'un contrat ou la conséquence d'un acte délictueux, y compris par négligence. Toute action contre National Instruments doit être intentée dans l'année qui suit la cause de cette action. National Instruments ne pourra être tenu responsable de tout retard en performance dû à des causes qui iraient au-delà de ce qu'il lui est raisonnablement possible de faire. La garantie fournie ici ne couvre pas les dommages, défauts, dysfonctionnements ou défauts de service dus à des erreurs commises par l'utilisateur dans l'interprétation des instructions de National Instruments en ce qui concerne l'installation, le fonctionnement et la maintenance. Elle ne couvre pas non plus les négligences, les modifications ou mauvais usages du produit de la part de l'utilisateur, les chutes de tension ou les surtensions, le feu, les inondations, les accidents, les agissements de tierces personnes, et tout autre événement incontrôlable.

Copyright

Conformément à la loi sur les droits d'auteur, ce document ne peut être ni reproduit ni transmis, sous aucune forme que ce soit, informatique ou mécanique, notamment par photocopie, enregistrement, stockage dans un système d'archivage de documentation, ni traduit intégralement ou en partie, sans l'autorisation écrite de National Instruments Corporation.

Marques déposées

BridgeVIEW™, CVI™, LabVIEW™, natinst.com™, National Instruments™, et NI-DAQ™ sont des marques de National Instruments Corporation.

Les noms des produits et des sociétés listés sont des marques ou des marques déposées de leurs sociétés respectives.

MISE EN GARDE CONCERNANT L'UTILISATION DES PRODUITS DE National Instruments DANS LES APPLICATIONS MÉDICALES ET HOSPITALIÈRES

Les produits de National Instruments ne sont pas conçus avec des composants et suivant des méthodes de tests prévus pour assurer un niveau de fiabilité convenant à leur utilisation dans les applications de traitement et de diagnostic sur les personnes. Les applications des produits de National Instruments impliquant des traitements médicaux ou cliniques peuvent potentiellement occasionner des blessures accidentelles à cause d'une panne des produits, ou à cause d'une erreur de la part de l'utilisateur ou du concepteur de l'application. Toute utilisation ou application des produits National Instruments pour ou dans des traitements médicaux ou cliniques doit être effectuée par un personnel médical correctement formé et qualifié, et toutes les garanties médicales d'usage, tous les équipements et toutes les procédures qui sont appropriées à cette situation particulière pour éviter les blessures graves ou la mort, doivent toujours être mis en œuvre lorsque l'on utilise les produits de National Instruments. Les produits de National Instruments N'ONT PAS été conçus pour se substituer à toute forme de procédé, procédure ou équipement utilisée pour la surveillance médicale ou pour garantir la santé publique dans les traitements médicaux ou cliniques.

Table des matières

Avant-propos

Organisation de ce manuel.....	xxv
Partie I : Concepts fondamentaux du G.....	xxv
Partie II : Objets de la face-avant	xxvi
Partie III : Programmation par diagramme.....	xxvii
Partie IV : Sujets avancés	xxviii
Annexes, glossaire et index	xxviii
Conventions utilisées dans ce manuel	xxix
Références bibliographiques.....	xxx
Communication avec l'utilisateur.....	xxx

PARTIE I

Concepts fondamentaux du G

Chapitre 1

Introduction à la programmation en G

Qu'est-ce que le G ?.....	1-1
Composantes d'un VI	1-1
Face-avant.....	1-3
Diagramme	1-4
Icône et connecteur.....	1-6
Aide.....	1-7
Aide pour la face-avant	1-7
Aide pour le diagramme	1-8
Aide pour les attributs	1-9
Référence en ligne	1-10

Chapitre 2

Edition des VIs

L'environnement graphique.....	2-1
Palette d'outils	2-4
Utilisation des menus	2-6
Menus locaux	2-6
Edition des VIs.....	2-7
Sélection des objets	2-7
Glisser-déposer des VIs, des images et du texte.....	2-8

Positionnement d'objets	2-9
Commande Réorganiser	2-10
Alignement d'objets	2-11
Répartition des objets	2-12
Copie des objets	2-13
Suppression d'objets	2-14
Nom des objets	2-14
Sous-titres de faces-avant	2-14
Étiquettes libres	2-16
Caractéristiques du texte	2-18
Redimensionnement des objets	2-22
Étiquettes	2-23
Ajout d'espace de travail	2-24
Coloriage des objets	2-24
Annulation	2-26
Création de descriptions d'objets	2-26
Création de descriptions pour les VIs	2-27
Enregistrement des VIs	2-28
Fichiers individuels	2-28
Bibliothèques de VIs (.LLB)	2-30
Création de bibliothèques de VIs	2-31
Enregistrement dans des bibliothèques de VIs existantes	2-32
Modification du contenu des bibliothèques	2-32

Chapitre 3

Utilisation des sous-VIs

Conception hiérarchique	3-1
Création de sous-VIs à partir de VIs	3-2
Création de l'icône	3-2
Définition des terminaux des connecteurs	3-4
Sélection et modification de terminaux	3-6
Affectation de terminaux aux commandes et indicateurs	3-7
Connexions nécessaires, recommandées et optionnelles pour les sous-VIs	3-8
Suppression des connexions de terminaux	3-10
Confirmation des connexions de terminaux	3-10
Création de sous-VIs à partir d'une partie d'un VI	3-11
Règles et recommandations	3-12
Nombre de connexions	3-12
Cycles	3-12
Attribute Nodes dans les boucles	3-13
Sélections illogiques	3-13

Variables locales et terminaux de face-avant	
dans les boucles	3-13
Structures Condition contenant des Attribute Nodes,	
variables locales ou terminaux de face-avant	3-14
Utilisation de la fenêtre Hiérarchie	3-15
Ouverture de la fenêtre Hiérarchie	3-15
Options de la fenêtre Hiérarchie.....	3-17
Options du menu Afficher.....	3-17
Boutons de barre d'outils de hiérarchie	3-18
Menu local de nœud dans la fenêtre Hiérarchie.....	3-19
Utilisation de la souris.....	3-20
Recherche de VIs dans la fenêtre Hiérarchie	3-21
Recherche de VIs, d'objets et de texte.....	3-22
Boîte de dialogue Rechercher.....	3-23
Recherche de VIs et d'autres objets	3-23
Recherche de texte	3-25
Options de recherche	3-25
Options de recherche de texte.....	3-26
Réduction de l'étendue de la recherche	3-27
Fenêtre Résultat de la recherche	3-27
Recherche d'éléments de recherche précédents	
ou suivants	3-28
Menu local de recherche pour les variables globales et locales et les	
Attribute Nodes.....	3-29

Chapitre 4

Exécution et mise au point des VIs et sous-VIs

Exécution des VIs	4-1
Exécution des VIs.....	4-1
Arrêt des VIs.....	4-4
Exécution répétée des VIs	4-4
Enregistrement des données de la face-avant	4-4
Récupération de données contrôlée par programme	4-7
Accès aux bases de données.....	4-7
Récupération de données à l'aide de fonctions d'E/S de fichier.....	4-8
Mise au point des VIs	4-9
Réparation des VIs invalides	4-9
Interprétation des messages d'erreur.....	4-11
Mise au point des VIs exécutables	4-14
Comprendre les mises en garde	4-16
Reconnaître des données non définies	4-16
Correction des erreurs de dépassement de gamme des VIs.....	4-17

Fonctions de mise au point.....	4-18
Exécution des VIs en mode Pas à pas.....	4-19
Exemple d'exécution d'un VI en mode Pas à pas.....	4-20
Utilisation des boutons Pas à pas.....	4-21
Lecture des chaînes d'appel.....	4-21
Exécution en mode Animation	4-22
Utilisation de l'outil Sonde.....	4-24
Création de sonde.....	4-26
Mise en place de points d'arrêt.....	4-27
Interruption de l'exécution	4-29
Comment reconnaître une interruption automatique.....	4-30
Utilisation des boutons de la barre d'outils lorsque	
l'exécution des sous-VIs est interrompue	4-31
Visualisation de la fenêtre Hiérarchie pendant	
l'interruption.....	4-31
Désactivation des fonctions de mise au point.....	4-31
Désactivation de sections de diagrammes.....	4-32

Chapitre 5

Impression et documentation des VIs

Impression	5-1
Configuration de l'impression	5-2
Impression en PostScript	5-2
Impression de la fenêtre active.....	5-2
Documentation des VIs	5-3
Impression de la documentation	5-3
Définition des formats d'impression	5-4
Définition d'autres options d'impression	5-5
Création de paramètres d'impression personnalisée.....	5-6
Impression contrôlée par programme	5-7
Contrôle du moment auquel ont lieu les impressions	5-8
Affinage des impressions.....	5-9
Définition de la mise en page	5-9
Autres méthodes d'impression.....	5-10
Imprimer/Exporter des descriptions de commandes et de VIs vers	
un fichier RTF ou HTML.....	5-10
Fichiers RTF et HTML contrôlés par programme.....	5-15
Localisation	5-15
Création de vos propres fichiers d'aide.....	5-15

Chapitre 6

Configuration des VIs et sous-VIs

Création de panneaux locaux	6-1
Options de configuration des VIs	6-2
Options d'exécution.....	6-2
Options des fenêtres	6-3
Options de documentation.....	6-6
Boîte de dialogue Configuration du nœud du sous-VI	6-7
Personnalisation de la barre de menus	6-8
Editeur de menus	6-9
Options de menu de l'éditeur de menus	6-12
Options de la barre d'outils de l'éditeur de menus.....	6-13
Gestion de la sélection des menus	6-14
Fonctions de gestion de la sélection de menu	6-15
Élément de menus sélectionné.....	6-15
Activer le repérage des menus	6-16
Fonctions de menu dynamiques.....	6-16
Insérer des éléments de menus.....	6-16
Supprimer des éléments de menus.....	6-18
Obtenir des infos sur un élément de menus	6-19
Définir les infos de l'élément de menus	6-19
Infos sur un raccourci de menus	6-20
Tags d'Élément d'application.....	6-20

Chapitre 7

Personnalisation de votre environnement

Paramètres des préférences	7-1
Préférences des chemins	7-2
Répertoires de la bibliothèque, répertoire temporaire, répertoire par défaut et répertoire des menus.....	7-3
Chemin de recherche du VI.....	7-4
Préférences concernant le disque et les performances	7-6
Préférences de la face-avant.....	7-10
Préférences pour le diagramme.....	7-12
Préférences concernant la mise au point.....	7-13
Préférences des couleurs	7-14
Préférences de fonte	7-16
Préférences d'impression	7-17
Préférences de l'historique	7-18
Préférences pour l'affichage de la date et de l'heure	7-22
Préférences diverses.....	7-23

Serveur : configuration.....	7-24
Serveur : accès TCP/IP	7-26
Serveur : VIs exportés.....	7-28
Comment sont stockées les préférences ?	7-31
Annuler	7-33
Personnalisation des palettes Commandes et Fonctions	7-34
Ajout de VIs et de commandes aux fichiers user.lib et instr.lib	7-34
Installation et modification des affichages de palettes.....	7-35
Editeur de palettes.....	7-35
Création de sous-palettes	7-36
Déplacement de sous-palettes.....	7-38
Fonctionnement des affichages des palettes	7-38

PARTIE II

Objets de la face-avant

Chapitre 8

Introduction aux objets de la face-avant

Construction de la face-avant	8-1
Options des commandes et indicateurs de la face-avant.....	8-2
Remplacement des commandes.....	8-4
Option d'utilisation du clavier pour les commandes	8-6
Option d'ordonnancement de la face-avant	8-8
Personnalisation des commandes de boîte de dialogue	8-9
Personnalisation des commandes avec des graphiques importés.....	8-10

Chapitre 9

Commandes et indicateurs numériques

Commandes et indicateurs numériques	9-2
Options des valeurs numériques	9-4
Afficher des entiers dans d'autres bases.....	9-4
Changer la représentation des valeurs numériques.....	9-5
Options de gamme des commandes et indicateurs numériques	9-6
Contrôle des gammes numériques	9-7
Format et précision des affichages numériques	9-9

Commandes et indicateurs numériques à glissière	9-12
Echelle de la glissière	9-14
Repères d'échelle	9-15
Changer les limites d'une échelle	9-15
Sélectionner une distribution non uniforme de repères d'échelle	9-16
Echelle du texte	9-18
Glissières pleines et à valeurs multiples	9-19
Commandes et indicateurs numériques rotatifs	9-21
Boîte de couleurs	9-23
Rampe de couleurs	9-24
Types d'unités	9-25
Entrée des unités	9-30
Unités et contrôle de type strict	9-31
Unités polymorphes	9-32

Chapitre 10

Commandes et indicateurs booléens

Création et utilisation de commandes et indicateurs booléens	10-1
Configuration des commandes et indicateurs booléens	10-3
Changer les étiquettes des booléens	10-3
Contrôle de la gamme de données des booléens	10-4
Configuration de l'action mécanique des commandes booléennes	10-4
Personnalisation des booléens avec des images importées	10-6

Chapitre 11

Commandes et indicateurs de chaîne de caractères

Utilisation des commandes et indicateurs de chaîne de caractères	11-1
Éléments de menu des commandes et indicateurs de chaîne de caractères	11-3
Éléments de menu Barre de défilement	11-3
Types d'affichage	11-4
Affichage normal	11-4
Affichage des codes à barre oblique inverse ('\'')	11-4
Affichage style mot de passe	11-6
Affichage hexadécimal	11-6
Limiter à une seule ligne	11-6
Mettre à jour la valeur en cours de frappe	11-7
Tables	11-7
Redimensionnement des tables, des rangées et des colonnes	11-8
Entrer et sélectionner des tables de données	11-8

Chapitre 12

Commandes et indicateurs de chemin et refnum

Commandes et indicateurs de chemin	12-1
Commandes et indicateurs de refnum	12-3

Chapitre 13

Commandes et indicateurs de type liste et menu déroulants

Commandes à liste déroulante	13-2
Création d'une liste d'éléments	13-2
Sélection des éléments d'une liste déroulante	13-3
Types de données des listes déroulantes	13-3
Éléments du menu local de liste déroulante	13-4
Visualiser	13-4
Mode de sélection	13-5
Mode du clavier	13-6
Désactiver un élément	13-7
Symbole d'un élément et ligne de division	13-7
Commandes de type menu déroulant	13-8
Ajouter des éléments de texte à des menus déroulants	13-10
Ajouter des éléments d'images aux menus déroulants	13-12
Changer la taille et le texte des menus déroulants texte & images	13-12
Commandes de type énuméré	13-13

Chapitre 14

Commandes et indicateurs de tableau et de cluster

Tableaux	14-2
Création de commandes de tableau	14-5
Dimensions du tableau	14-8
Affichage de l'indice du tableau	14-9
Affichage de tableaux sous forme élément unique ou tabulaire	14-10
Manipulation des tableaux	14-11
Taille et valeurs par défaut des tableaux	14-11
Éléments du tableau	14-13
Recherche de la taille des tableaux	14-14
Déplacement ou redimensionnement des tableaux	14-14
Sélection des cellules du tableau	14-15
Un exemple de sélection de cellules d'un tableau	14-15
Tableaux G et tableaux d'autres systèmes	14-17

Clusters	14-20
Création des clusters	14-21
Manipulation et configuration des éléments du cluster	14-22
Éléments de cluster	14-22
Valeurs par défaut du cluster	14-22
Ordonnancement des éléments du cluster	14-23
Déplacement ou redimensionnement des clusters	14-24
Assemblage des clusters	14-25
Fonction Assembler	14-25
Fonction “Assembler par nom”	14-26
Fonction “Tableau en cluster”	14-28
Séparation des clusters	14-30
Fonction Séparer	14-30
Fonction “Séparer par nom”	14-30
Fonction “Cluster en tableau”	14-32
Remplacement d’éléments de clusters	14-33

Chapitre 15

Indicateurs et commandes de graphe et de graphe déroulant

Création de forme d’onde et de graphes XY	15-2
Graphes à tracé unique	15-3
Types de données des graphes oscilloscopiques	15-3
Types de données des graphes XY	15-4
Graphe déroulant multicourbes	15-5
Types de données des graphes oscilloscopiques	15-5
Types de données des graphes XY	15-10
Définir des options personnalisées sur un graphe	15-11
Options d’échelle	15-12
Espace des repères	15-13
Formatage	15-14
Mise à l’échelle automatique	15-17
Ajuster l’échelle	15-17
Options de défilement et de zoom	15-17
Options de légende	15-19
Graphes déroulants	15-22
Types de données des graphes déroulants	15-22
Options des graphes déroulants	15-24
Modes de mise à jour des graphes déroulants	15-25
Tracés empilés et tracés superposés	15-27
Curseurs du graphe	15-29

Graphes déroulants d'intensité	15-34
Options des graphes déroulants d'intensité.....	15-36
Représentation des couleurs	15-38
Graphes d'intensité.....	15-39
Type de données des graphes d'intensité.....	15-39
Options des graphes d'intensité	15-39

Chapitre 16

Commandes ActiveX

Éléments ActiveX de la face-avant	16-1
Commande et indicateur de variable ActiveX.....	16-1
Container ActiveX.....	16-2
Construire des palettes ActiveX.....	16-5

PARTIE III

Programmation par diagramme

Chapitre 17

Introduction au diagramme

Terminaux et nœuds	17-1
Terminaux	17-1
Terminaux de commande et d'indicateur	17-2
Constantes.....	17-3
Constantes définies par l'utilisateur.....	17-4
Constantes universelles	17-7
Nœuds	17-8
Fonctions	17-8
Structures	17-11
Remplacement et insertion des objets du diagramme	17-12
Ajout automatique de constantes, commandes et indicateurs	17-13

Chapitre 18

Câblage du diagramme

Techniques de câblage.....	18-1
Adaptation des liaisons	18-5
Sélection, déplacement et suppression des fils de liaison.....	18-6
Câblage hors écran	18-8

Résolution des problèmes de câblage	18-8
Fils incorrects	18-8
Conflits de type pour les fils de liaison	18-9
Plusieurs sources de fil de liaison	18-10
Absence de source de liaison	18-11
Segments déconnectés	18-11
Cycles de fils de liaison	18-12
Situations de câblage à éviter	18-12
Fils de liaison en boucle	18-13
Portions de fil masquées	18-13
Câblage sous des objets	18-14

Chapitre 19

Structures

Structures de boucle For et de boucle While	19-3
Boucle For	19-3
Boucle While	19-4
Placement d'objets à l'intérieur des structures	19-4
Placement et dimensionnement de structures sur le diagramme	19-5
Placement de terminaux à l'intérieur des boucles	19-6
Auto-indexation	19-7
Auto-indexation pour définir le nombre d'itérations de la	
boucle For	19-8
Auto-indexation des boucles While	19-8
Exécution d'une boucle For zéro fois	19-9
Registres à décalage	19-9
Structures Condition et Séquence	19-12
Structures Condition	19-13
Structures Séquence	19-18
Edition des structures Séquence et Condition	19-20
Déplacement entre les sous-diagrammes	19-20
Ajout de sous-diagrammes	19-21
Suppression de sous-diagrammes	19-22
Problèmes de câblage de structure	19-22
Affectation de plusieurs valeurs à une variable locale	
de séquence	19-22
Tunnel non câblé dans toutes les conditions d'une structure	
Condition	19-23
Chevauchement de tunnels	19-23
Câblage à partir de plusieurs étapes d'une structure Séquence	19-24

Câblage par-dessous plutôt qu'à travers une structure	19-25
Suppression de structures sans effacer les éléments qui se trouvent à l'intérieur	19-26

Chapitre 20

Boîtes de calcul

Utilisation des Boîtes de calcul	20-1
Fonctions et opérateurs de la Boîte de calcul	20-4
Syntaxe de la Boîte de calcul	20-7
Erreurs de la Boîte de calcul	20-9

Chapitre 21

VI Serveur

Utilisation du VI Serveur	21-1
Fonctions du VI Serveur	21-3
Références d'application et de VI	21-4
Utilisation des nœuds de propriété et de méthode avec les références d'application et de VI	21-4
Exemple de propriétés et méthodes de classe de VI et d'application	21-7
Manipulation des méthodes et propriétés de classe de VI	21-7
Manipulation des méthodes et des propriétés de classe d'application	21-9
Manipulation des méthodes et des propriétés de classe d'application et de VI	21-10
Refnums de VI de type strict	21-10
Exemple de refnums de VI de type strict	21-11

Chapitre 22

Attribute Nodes

Création d'attribut nodes	22-1
Utilisation de l'aide d'attribut node	22-6
Attributs de base	22-6
Attribut Visible	22-6
Attribut Désactiver	22-7
Attribut "Focus de touche"	22-7
Attribut Clignotement	22-8
Attribut Position	22-8
Attribut Limites (lecture seule)	22-9
Exemples d'attributs spécifiques à certaines commandes ou indicateurs	22-9
Modification de la couleur du tracé sur un graphe déroulant	22-9
Définition des chaînes de caractères d'un attribut de booléen	22-10
Définition des chaînes de caractères des commandes de type menu déroulant	22-11

Utilisation d'un élément de liste déroulante à double-clic.....	22-12
Présentation d'options aux utilisateurs de manière sélective.....	22-13
Lecture par programme des curseurs du graphe	22-13

Chapitre 23

Variables globales et locales

Variables globales.....	23-1
Variables locales.....	23-4

PARTIE IV

Sujets avancés

Chapitre 24

Commandes personnalisées et définitions de type

Commandes personnalisées	24-2
Utilisation de l'éditeur de commandes	24-2
Application des modifications d'une commande personnalisée	24-4
Commandes personnalisées valides	24-4
Enregistrement des commandes personnalisées	24-5
Utilisation des commandes personnalisées	24-5
Création d'icônes.....	24-5
Instances indépendantes des commandes personnalisées.....	24-6
Option Mode Personnalisé.....	24-6
Composantes indépendantes.....	24-7
Fenêtre des composantes de l'éditeur de commandes.....	24-8
Menus locaux du mode Personnalisé pour les différentes composantes.....	24-10
Composantes décoratives	24-10
Composantes décoratives à plusieurs images	24-12
Composantes décoratives à images indépendantes.....	24-14
Composantes de texte.....	24-15
Commandes composantes	24-16
Ajout de composantes décoratives à des commandes personnalisées.....	24-17
Avertissements à propos des commandes personnalisées.....	24-18
Définitions de type.....	24-19
Définition de type général : correspondance des types de données	24-19
Définition de type strict : tout doit correspondre.....	24-20
Définitions de type sur le diagramme.....	24-20
Création de définitions de type.....	24-20
Utilisation des définitions de type	24-21

Mise à jour des définitions de type	24-22
Recherche des définitions de type.....	24-23
Définitions de type cluster	24-23

Chapitre 25

Appel de code d'autres langages

Exécution d'autres applications à partir de vos VIs	25-1
Utilisation du nœud "Appeler une fonction d'une DLL"	25-2
Utilisation des Code Interface Nodes.....	25-3
Appeler une fonction d'une	25-3
Conventions d'appel (Windows)	25-5
Listes des paramètres	25-5
Appel de fonctions qui attendent d'autres types de données	25-8
Convertisseur de panneau de fonction LabWindows/CVI.....	25-9
Processus de conversion.....	25-10

Chapitre 26

Comprendre le système d'exécution du G

Vue d'ensemble du multitâche	26-1
Multithreading.....	26-2
Le système d'exécution du G.....	26-3
Système d'exécution de base	26-3
Gestion de l'interface utilisateur dans une application à un seul thread.....	26-4
Application à plusieurs threads et systèmes d'exécution multiples	26-4
Nœuds synchrones/de blocage	26-7
Priorité des tâches.....	26-7
Priorité des tâches à l'aide des fonctions Attendre	26-8
Priorité dans la configuration du VI.....	26-8
Priorités du thread de l'interface utilisateur et du système d'exécution à un seul thread	26-9
Priorités dans les autres systèmes d'exécution du système à plusieurs threads.....	26-10
Niveau de priorité de type sous-programme.....	26-11
Vue d'ensemble de l'exécution ré-entrante	26-12
Exemples d'utilisation du mode ré-entrant	26-14
Utilisation d'un VI d'attente.....	26-14
Utilisation d'un VI de stockage qui n'est pas censé partager ses données	26-15
Synchronisation de l'accès aux variables globales, aux variables locales et aux ressources externes	26-16
Conditions de compétition.....	26-16
Variables globales fonctionnelles	26-18

Sémaphores.....	26-19
Autres fonctions de synchronisation	26-21
Conseils d'utilisation des systèmes d'exécution et des priorités.....	26-22

Chapitre 27

Gestion de vos applications

Arrangement des fichiers à l'aide des bibliothèques de VIs.....	27-1
Sauvegarde de vos fichiers	27-2
Distribution des VIs	27-2
VIs protégés par un mot de passe	27-4
Boîte de dialogue "Enregistrer avec options"	27-6
Concevoir des applications avec plusieurs développeurs	27-8
Conserver des copies maîtresses	27-9
Fenêtre Historique du VI.....	27-9
Numéros de révision	27-11
Réinitialisation des informations de l'historique	27-12
Impression des informations de l'historique	27-13
Paramétrage des options associées dans les boîtes de dialogue	
Configuration du VI et Préférences.....	27-13

Chapitre 28

Performances

Optimisation des performances des VIs	28-1
Visualisation des résultats	28-3
Informations temporelles	28-4
Informations sur la mémoire	28-5
Vitesse d'exécution des VIs.....	28-6
Entrée/Sortie	28-7
Affichage à l'écran	28-8
Autres questions	28-9
Diagrammes parallèles	28-9
Temps machine d'accès aux sous-VIs	28-11
Calculs superflus dans des boucles	28-11
Utilisation de la mémoire par les VIs	28-13
Mémoire virtuelle	28-14
Mémoire du Macintosh.....	28-14
Gestion de la mémoire par les composantes des VIs	28-15
Programmation par flux de données et buffers de données.....	28-16
Contrôle de l'utilisation de la mémoire	28-18
Règles pour une meilleure utilisation de la mémoire	28-19
Gestion de mémoire dans les faces-avant	28-20
Les sous-VIs peuvent réutiliser la mémoire des données	28-22

Les variables locales ne peuvent pas réutiliser la mémoire des données	28-22
Les variables globales gardent toujours des copies de leurs données	28-23
Comprendre à quel moment la mémoire est libérée	28-23
Déterminer à quel moment les sorties peuvent réutiliser les buffers d'entrée.....	28-24
Types de données compatibles	28-24
Comment générer des données du type voulu.....	28-25
Eviter de redimensionner constamment les données.....	28-26
Exemple 1 : construction de tableaux	28-27
Exemple 2 : recherche dans les chaînes de caractères	28-29
Développement de structures de données efficaces	28-32
Etude de cas 1 : éviter les types de données compliqués	28-34
Etude de cas 2 : table globale de types de données mélangés.....	28-35
Approche évidente	28-36
Approche alternative n° 1	28-37
Approche alternative n° 2	28-37
Etude de cas 3 : une table de chaînes de caractères globale statique	28-39

Chapitre 29

Portabilité et localisation

VIs portables et non portables	29-1
Port d'une plate-forme à une autre	29-2
Différences entre les caractères séparateurs.....	29-2
Différences de résolution et de fonte	29-3
Chevauchement d'étiquettes	29-4
Différences entre les images	29-5
Localisation des VIs	29-6
Importer et exporter les chaînes de caractères des VIs	29-7
Syntaxe du fichier de chaînes de caractères du VI.....	29-7
Edition du titre de la fenêtre du VI	29-13
Point et virgule comme séparateurs décimaux.....	29-14
Formater la chaîne de caractères de la date/heure.....	29-15
Port d'une application en langage G à une autre	29-15
De LabVIEW à BridgeVIEW	29-15
De BridgeVIEW à LabVIEW	29-16

Annexe A

Formats d'enregistrement des données

Formats des données pour les commandes et indicateurs de la face-avant	A-1
Booléens	A-1
Nombres	A-1
Précision étendue	A-1
Double précision	A-3
Simple précision.....	A-3
Entier long.....	A-3
Entier mot.....	A-3
Entier octet	A-3
Tableaux	A-4
Chaînes de caractères	A-5
Chemins.....	A-5
Clusters.....	A-5
Descripteurs de type.....	A-7
Types de données	A-8
Tableau.....	A-11
Cluster.....	A-12
Données aplaties	A-12
Scalaire.....	A-13
Chaînes de caractères, handles et chemins	A-14
Tableaux	A-14
Clusters.....	A-15

Annexe B

Questions courantes concernant le G

Graphes et graphes déroulants	B-1
Messages d'erreur et blocages	B-5
Problèmes de plate-forme et compatibilité	B-6
Impression.....	B-8
Divers.....	B-11

Annexe C

Informations à l'attention du client

Glossaire

Index

Figures

Figure 2-1.	Boîte de dialogue Etiquette de la face-avant	2-15
Figure 2-2.	Menu local d'une étiquette	2-15
Figure 2-3.	Objet sur le diagramme, avec le sous-titre visible	2-15
Figure 2-4.	Boîte de dialogue d'aide contextuelle	2-16
Figure 5-1.	Boîte de dialogue Imprimer la documentation.....	5-11
Figure 5-2.	Boîte de dialogue Imprimer la documentation, Fichier RTF	5-12
Figure 5-3.	Boîte de dialogue Imprimer la documentation, Fichier HTML	5-13
Figure 6-1.	Editeur de menus	6-9
Figure 6-2.	Choix d'un Elément d'application	6-11
Figure 6-3.	Fonction "Elément de menus sélectionné".....	6-14
Figure 6-4.	Activer le repérage des menus	6-15
Figure 6-5.	Insertion de menu dynamique	6-18
Figure 14-1.	Ouvrez un menu local dans l'affichage d'élément et sélectionnez la commande Booléen	14-6
Figure 14-2.	Relâchez le bouton de la souris. Le menu disparaît et le curseur setransforme en icône Défilement des fenêtres	14-7
Figure 14-3.	Faites glisser l'icône Défilement des fenêtres dans l'affichage d'élément.....	14-7
Figure 14-4.	Cliquez sur le bouton de la souris. La commande est alors déposée dans l'affichage.	14-7
Figure 19-1.	Incohérence du type de la structure Condition.....	19-14
Figure 19-2.	Eléments du menu local d'une structure Condition	19-16
Figure 19-3.	Boîte de dialogue "Réorganiser les conditions".....	19-17
Figure 29-1.	Exemple de localisation de VI	29-6

Tableaux

Tableau 4-1.	Messages d'erreur.....	4-11
Tableau 4-2.	Mise en place de points d'arrêt.....	4-28
Tableau 5-1.	Fichiers JPEG résultants.....	5-14
Tableau 5-2.	Exemples de noms de fichiers d'images.....	5-15
Tableau 6-1.	Tags d'Élément d'application.....	6-20
Tableau 7-1.	Serveur : accès TCP/IP.....	7-28
Tableau 7-2.	Serveur : Entrées de liste de VI exporté.....	7-30
Tableau 9-1.	Options de gamme des nombres à virgule flottante.....	9-6
Tableau 9-2.	Unités de base.....	9-27
Tableau 9-3.	Unités dérivées à noms spéciaux.....	9-27
Tableau 9-4.	Unités supplémentaires en usage avec des unités SI.....	9-28
Tableau 9-5.	Unités CGS.....	9-29
Tableau 9-6.	Autres unités.....	9-29
Tableau 11-1.	Codes '\` du G.....	11-4
Tableau 17-1.	Symboles de terminaux de commande et d'indicateur du G.....	17-2
Tableau 20-1.	Fonctions de la Boîte de calcul.....	20-4
Tableau 20-2.	Erreurs de la Boîte de calcul.....	20-9
Tableau 29-1.	Description des tags des VIs.....	29-8
Tableau 29-2.	Contenu de la face-avant.....	29-9
Tableau 29-3.	Tags pour la [commande].....	29-10
Tableau 29-4.	Valeurs par défaut pour les chaînes de caractères.....	29-11
Tableau 29-5.	Description des tags pour les cellules des tables, les noms des tracés du graphe et les noms des curseurs.....	29-12
Tableau A-1.	Types de données numériques scalaires.....	A-8
Tableau A-2.	Types de données non numériques.....	A-9

Avant-propos

Le *Manuel de référence de programmation en G* décrit comment créer, éditer et exécuter des instruments virtuels (VIs) utilisant le langage de programmation G. Ce manuel explique la face-avant et le diagramme, les commandes et indicateurs numériques, booléens, de chaînes de caractères, tableaux et clusters, les graphes, les graphes déroulants et le câblage du diagramme.

Organisation de ce manuel

Ce manuel aborde quatre domaines différents. Les chapitres 1 à 7 introduisent les concepts fondamentaux du G. Les chapitres 8 à 16 expliquent comment utiliser les objets de la face-avant. Les chapitres 17 à 23 expliquent les objets et techniques de programmation par diagramme, et les chapitres 24 à 27 traitent des sujets avancés concernant le G.

Partie I : Concepts fondamentaux du G

Cette section contient des informations de base sur la programmation en G, la création, l'édition et la personnalisation des instruments virtuels (VIs).

La Partie I, *Concepts fondamentaux du G*, contient les chapitres suivants :

- Le chapitre 1, *Introduction à la programmation en G*, décrit l'approche unique du G à la programmation. Il explique également comment commencer à utiliser le G pour développer des programmes.
- Le chapitre 2, *Edition des VIs*, décrit les fonctions de base de la construction et de l'utilisation des VIs et contient des informations sur les palettes et menus. Il décrit également des tâches élémentaires telles que la manière de créer des objets, la manière de changer d'outil, et la manière d'ouvrir, d'exécuter et d'enregistrer des VIs.
- Le chapitre 3, *Utilisation des sous-VIs*, décrit le concept de la conception hiérarchique des applications G et présente deux méthodes de création de sous-VIs. Ce chapitre décrit également deux utilitaires — la fenêtre Hiérarchie, qui affiche la hiérarchie de vos VIs et l'utilitaire Chercher, qui recherche les occurrences de sous-VIs, de même que tous les autres objets ou chaînes de texte indiqués.
- Le chapitre 4, *Exécution et mise au point des VIs et sous-VIs*, décrit le fonctionnement et la mise au point des VIs et configuration des VIs et des sous-VIs pour des modes d'exécution spéciaux.

- Le chapitre 5, *Impression et documentation des VIs*, aborde plusieurs questions concernant l'impression et la documentation des VIs.
- Le chapitre 6, *Configuration des VIs et sous-VIs*, explique comment personnaliser le comportement des VIs par le biais des boîtes de dialogue **Configuration du VI** et **Configuration du nœud du sous-VI**.
- Le chapitre 7, *Personnalisation de votre environnement*, explique comment personnaliser votre environnement en paramétrant les préférences de l'éditeur pour des fonctions telles que l'impression, l'affichage, et **Annuler**, et en modifiant le contenu des palettes **Commandes** et **Fonctions**.

Partie II : Objets de la face-avant

Cette section contient des informations sur les objets de la face-avant, les commandes et indicateurs et les commandes ActiveX.

La Partie II, *Objets de la face-avant*, contient les chapitres suivants :

- Le chapitre 8, *Introduction aux objets de la face-avant*, présente la face-avant et ses deux composantes : les commandes et les indicateurs. Il vous explique également comment importer des graphiques provenant d'autres applications afin de les utiliser dans vos commandes.
- Le chapitre 9, *Commandes et indicateurs numériques*, décrit comment créer, utiliser et configurer les divers styles de commandes et indicateurs numériques.
- Le chapitre 10, *Commandes et indicateurs booléens*, décrit comment créer, utiliser et configurer les commandes et indicateurs booléens.
- Le chapitre 11, *Commandes et indicateurs de chaîne de caractères*, décrit comment utiliser les commandes et indicateurs de chaîne de caractères, ainsi que la table. Vous pouvez accéder à ces objets au moyen de la palette **Commandes»Chaîne de caractères & table**.
- Le chapitre 12, *Commandes et indicateurs de chemin et refnum*, décrit comment utiliser les commandes de chemin de fichier et les refnums, qui sont disponibles à partir de la palette **Commandes»Chemin & Refnum**.
- Le chapitre 13, *Commandes et indicateurs de type liste et menu déroulants*, décrit les commandes et indicateurs de type liste déroulante et de type menu déroulant, qui sont disponibles à partir de la palette **Commandes»Listes & menus déroulants**.

- Le chapitre 14, *Commandes et indicateurs de tableau et de cluster*, explique comment utiliser les tableaux et les clusters. Vous accédez aux tableaux et clusters à partir de la palette **Commandes» Tableaux et Clusters**.
- Le chapitre 15, *Indicateurs et commandes de graphe et de graphe déroulant*, explique comment créer et utiliser les indicateurs de graphes et de graphes déroulants dans la palette **Commandes» Graphe**.
- Le chapitre 16, *Commandes ActiveX*, décrit les possibilités du Container ActiveX, qui augmente les interactions entre les logiciels écrits en G et les autres applications.

Partie III : Programmation par diagramme

Cette section contient des informations sur les composants requis pour construire et manipuler un diagramme en G.

La Partie III, *Programmation par diagramme*, contient les chapitres suivants :

- Le chapitre 17, *Introduction au diagramme*, décrit les terminaux et les nœuds—deux des trois éléments que vous utilisez pour construire un diagramme. Le troisième élément, le câblage, est abordé dans le chapitre 18, *Câblage du diagramme*.
- Le chapitre 18, *Câblage du diagramme*, explique comment connecter des terminaux au diagramme en les reliant avec des fils de liaison.
- Le chapitre 19, *Structures*, explique comment utiliser la boucle For, la boucle While, la structure Condition et la structure Séquence. Ces structures se trouvent dans la palette **Fonctions»Structures**.
- Le chapitre 20, *Boîtes de calcul*, explique comment utiliser la boîte de calcul pour exécuter des opérations mathématiques sur le diagramme. La boîte de calcul est disponible à partir de la palette **Fonctions»Structures**.
- Le chapitre 21, *VI Serveur*, décrit le mécanisme permettant de contrôler les VIs et les applications au moyen d'un programme. Ce chapitre explique également comment contrôler les VIs ou les applications à distance.
- Le chapitre 22, *Attribute Nodes*, explique comment utiliser les attribute nodes pour définir et lire les attributs des commandes de la face-avant au moyen d'un programme. Parmi les attributs utiles figurent les couleurs d'affichage, la visibilité de la commande, les chaînes de caractères des menus pour une commande à menu

déroulant, les couleurs des tracés des graphes ou des graphes déroulants et les curseurs des graphes.

- Le chapitre 23, *Variables globales et locales*, explique comment définir et utiliser des variables globales et des variables locales. Utilisez des variables globales pour accéder facilement à un ensemble de valeurs particulier à partir de plusieurs VIs. Les variables locales remplissent un rôle similaire au sein d'un VI unique.

Partie IV : Sujets avancés

Cette section contient des informations sur les fonctions et techniques avancées du G que vous pouvez utiliser pour créer des instruments virtuels.

La Partie IV, *Sujets avancés*, contient les chapitres suivants :

- Le chapitre 24, *Commandes personnalisées et définitions de type*, introduit les commandes personnalisées et les définitions de type.
- Le chapitre 25, *Appel de code d'autres langages*, décrit diverses méthodes pour appeler du code écrit dans un autre langage.
- Le chapitre 26, *Comprendre le système d'exécution du G*, décrit le fonctionnement en multitâche et l'exécution des VIs.
- Le chapitre 27, *Gestion de vos applications*, décrit comment gérer les fichiers dans vos applications en G.
- Le chapitre 28, *Performances*, est divisé en trois sections. La première section décrit le Performance Profiler, une fonction qui vous donne des informations sur le temps d'exécution de vos VIs et contrôle les applications qui fonctionnent en mono-threading, multi-threading et sur plusieurs processeurs. La deuxième section décrit les facteurs qui affectent la vitesse d'exécution. La troisième section décrit les facteurs qui affectent l'usage de mémoire.
- Le chapitre 29, *Portabilité et localisation*, aborde les problèmes concernant le port de VIs d'une plate-forme à une autre et la localisation des VIs.

Annexes, glossaire et index



- L'annexe A, *Formats d'enregistrement des données*, décrit les formats sous lesquels vous pouvez enregistrer des données.
- L'annexe B, *Questions courantes concernant le G*, apporte des réponses à certaines des questions couramment posées par les utilisateurs du G.

- L'Annexe C, *Informations à l'attention du client*, contient des formulaires qui vous aident à rassembler les informations nécessaires pour aider National Instruments à résoudre vos problèmes techniques et un formulaire que vous pouvez utiliser pour apporter vos commentaires sur la documentation du produit.
- Le *Glossaire* contient une liste alphabétique et une description des termes utilisés dans ce manuel.
- L'*Index* contient une liste alphabétique des termes et sujets clés qui figurent dans ce manuel, accompagnés de la page où vous pouvez trouver chacun d'eux.

Conventions utilisées dans ce manuel

Les conventions suivantes sont utilisées dans ce manuel :

- <> Les crochets en pointe désignent une touche sur le clavier : par exemple, <maj>. Les crochets en pointe contenant des nombres séparés par deux points représentent une gamme de valeurs correspondant à un bit ou à un nom de signal : par exemple, DBIO<3..0>.
- [] Les crochets droits désignent des éléments optionnels : par exemple, [réponse].
- Un tiret entre deux ou plusieurs noms de touches à l'intérieur de crochets en pointe indique que vous devez appuyer simultanément sur les touches : par exemple, <Contrôle-Alt-Supprimer>.
- » Le symbole » vous indique les éléments de menu et les options de boîte de dialogue à sélectionner successivement jusqu'à une action finale. La séquence **Fichier»Mise en page»Options»Fontes de substitution** vous invite à ouvrir le menu **Fichier**, à sélectionner l'élément **Mise en page**, à sélectionner **Options**, et finalement à sélectionner l'option **Fontes de substitution** dans la dernière boîte de dialogue.
- ◆ Le symbole ◆ indique que le texte qui le suit s'applique uniquement à un produit, système d'exploitation ou version de logiciel spécifique.
- gras** Le texte en caractères gras désigne les noms de menus, éléments de menu, paramètres, boîtes de dialogue, boutons ou options de boîte de dialogue, icônes, fenêtres, onglets de Windows 95 ou LED.
- gras italique* Le texte en caractères gras et en italique désigne l'objectif d'une activité, une remarque, une mise en garde ou un avertissement.

<Contrôle>	Le nom des touches apparaît avec une majuscule.
<i>italique</i>	Le texte en italique désigne des variables, du texte d'une importance particulière, une référence croisée ou une introduction à un concept clé. Cette fonte désigne également du texte que vous devez remplacer par le mot ou la valeur approprié, comme dans Windows 3.x.
non proportionnel	Le texte qui apparaît dans cette fonte désigne du texte ou des caractères que vous devez entrez littéralement à partir du clavier, des sections de code, des exemples de programmation et des exemples de syntaxe. Cette fonte est également utilisée pour les noms des lecteurs de disques, les chemins, les répertoires, les programmes, les sous-programmes, les noms de périphériques, les fonctions, les opérations, les variables, les noms de fichiers et les extensions, ainsi que pour les instructions et commentaires provenant de programmes.
non proportionnel gras	Le texte en caractères gras qui apparaît dans cette fonte désigne les messages et les réponses que l'ordinateur affiche automatiquement à l'écran. Cette fonte sert aussi à mettre l'accent sur des lignes de code qui sont différentes des autres exemples.
<i>non proportionnel italique</i>	Le texte en italique qui apparaît dans cette fonte indique que vous devez entrer les mots ou valeurs appropriés à la place de ces éléments.
chemins	Dans ce manuel, les chemins sont notés en utilisant des barres obliques inverses (\) pour séparer les noms d'unité, les répertoires, les dossiers et les fichiers.
Plate-forme	Le texte qui apparaît dans cette fonte désigne des informations relatives à une plate-forme spécifique.
	Cette icône placée à la gauche de texte en caractères gras et en italique indique une remarque, ce qui souligne des informations importantes.
	Cette icône placée à la gauche de texte en caractères gras et en italique indique une mise en garde, ce qui vous informe des précautions à prendre pour éviter les blessures, la perte de données ou un blocage du système.
	Les abréviations, les sigles, les préfixes métriques, les noms mnémoniques, les symboles et les termes figurent dans le <i>Glossaire</i> .

Références bibliographiques

Les documents suivants contiennent des informations qui pourront vous être utiles lorsque vous lirez ce manuel :

- *Manuel de l'utilisateur LabVIEW*
- *BridgeVIEW User Manual*
- *Manuel de base d'acquisition de données LabVIEW*
- *Manuel de référence des VIs et des fonctions de LabVIEW*
- *Guide de prise en main de LabVIEW*
- *Référence en ligne de LabVIEW ou BridgeVIEW*, disponible en sélectionnant **Aide»Référence en ligne**
- *Tutorial en ligne LabVIEW (Windows uniquement)* que vous lancez à partir de la boîte de dialogue LabVIEW
- *Carte d'initiation LabVIEW*
- *Carte de référence rapide de programmation en G*
- *VXI VI Reference Manual*
- *LabVIEW Code Interface Reference*, disponible en format PDF (*portable document format*) sur le CD ou les disques de programme de votre logiciel.

Communication avec l'utilisateur

National Instruments aimerait recevoir vos commentaires sur ses produits et manuels. Les applications que vous développez avec nos produits nous intéressent et nous aimerions vous aider en cas de problèmes avec ces derniers. Pour que vous puissiez nous contacter facilement, ce manuel contient des formulaires de commentaires et de configuration que vous êtes invité à remplir. Vous trouverez ces formulaires dans l'Annexe C, *Informations à l'attention du client*, à la fin de ce manuel.

Partie I

Concepts fondamentaux du G

Cette section contient des informations de base concernant la programmation en G et la création, l'édition et la personnalisation des instruments virtuels (VIs).

La Partie I, *Concepts fondamentaux du G*, contient les chapitres suivants :

- Le chapitre 1, *Introduction à la programmation en G*, décrit l'approche unique du G à la programmation. Il explique également comment commencer à utiliser le G pour développer des programmes.
- Le chapitre 2, *Edition des VIs*, décrit les opérations de base pour la construction et l'utilisation des VIs et contient des informations sur les palettes et menus. Il décrit également des tâches élémentaires telles que la manière de créer des objets, de changer d'outil et d'exécuter enregistrer des VIs.
- Le chapitre 3, *Utilisation des sous-VIs*, décrit le concept de la conception hiérarchique des applications G et présente deux méthodes de création de sous-VIs. Ce chapitre décrit également deux utilitaires : la fenêtre Hiérarchie, qui affiche la hiérarchie de vos VIs et l'utilitaire Chercher, qui recherche les occurrences de sous-VIs, de même que tous les autres objets ou chaînes de texte indiqués.
- Le chapitre 4, *Exécution et mise au point des VIs et sous-VIs*, décrit le fonctionnement et la mise au point des VIs et explique la configuration des VIs et des sous-VIs pour des modes d'exécution spéciaux.
- Le chapitre 5, *Impression et documentation des VIs*, aborde plusieurs questions concernant l'impression et la documentation des VIs.
- Le chapitre 6, *Configuration des VIs et sous-VIs*, explique comment personnaliser le comportement des VIs par le biais des boîtes de dialogue **Configuration du VI** et **Configuration du nœud du sous-VI**.

- Le chapitre 7, *Personnalisation de votre environnement*, explique comment personnaliser votre environnement en paramétrant les préférences d'édition pour des fonctions telles que l'impression, l'affichage et **Annuler**, et en modifiant le contenu des palettes **Commandes** et **Fonctions**.

Introduction à la programmation en G

Ce chapitre présente l'approche unique avec laquelle le G aborde la programmation. Vous y trouverez également une explication des notions de base nécessaires au développement de programmes à l'aide du G.

Qu'est-ce que le G ?

Le G est le langage de programmation propre à LabVIEW. Il fait également partie intégrante de BridgeVIEW, l'environnement de développement d'applications de National Instruments.

Le G, tel que le C ou le Basic, est un système de programmation à usage général, disposant de bibliothèques de fonctions pour n'importe quelle tâche de programmation. Le G est accompagné de bibliothèques pour l'acquisition de données, le contrôle d'instruments GPIB et série, ainsi que l'analyse, la présentation et le stockage de données. Le G comprend également des outils conventionnels de développement vous permettant de définir des points d'arrêt, d'animer l'exécution pour observer la progression des données dans le programme et d'exécuter le programme pas à pas pour faciliter la mise au point et le développement.

Le G diffère des autres langages de programmation dans le sens suivant : alors que les autres langages de programmation sont des langages *textuels*, le G est un langage *graphique*.

Composantes d'un VI

Le G est un système de programmation à usage général, mais disposant également de bibliothèques de fonctions conçues spécifiquement pour l'acquisition de données et le contrôle d'instruments. Les programmes écrits en G sont appelés *VI*s (pour *Virtual Instruments : Instruments virtuels*) en raison de leur apparence et de leur fonctionnement semblables à ceux d'instruments réels. Ils sont cependant identiques aux fonctions des langages de programmation conventionnels.

Un VI comprend une interface utilisateur interactive (un diagramme équivalent au code source) et des connexions entre icônes définissant les VIs et leur permettant d'être appelés par des VIs d'un niveau supérieur. Les VIs sont structurés comme suit :

- L'interface utilisateur interactive d'un VI est appelée *face-avant*, en raison de sa ressemblance à la face-avant d'un instrument réel. La face-avant peut contenir des boutons rotatifs, des boutons-poussoirs, des graphiques et d'autres commandes et indicateurs. Vous pouvez entrer vos données à l'aide d'une souris ou d'un clavier et visualiser les résultats à l'écran.
- Le VI reçoit des instructions d'un *diagramme*, que vous éditez en G. Le diagramme est une solution graphique à un problème de programmation. Le diagramme est également le code source du VI.
- Les VIs suivent un format modulaire hiérarchique. Vous pouvez les utiliser comme programmes principaux ou comme sous-programmes au sein d'autres programmes ou sous-programmes. Un VI utilisé à l'intérieur d'un autre VI est appelé *sous-VI*. L'*icône* et le *connecteur* d'un VI fonctionnent comme une liste graphique de paramètres permettant aux autres VIs de transmettre des données à un sous-VI.

C'est grâce à ces fonctions que le G tire le meilleur parti du concept de *programmation modulaire*. Vous divisez une application en une série de tâches, que vous divisez à nouveau jusqu'à ce qu'une application compliquée devienne une série de tâches simples. Vous construisez un VI pour accomplir chaque sous-tâche et les combinez dans un autre diagramme pour accomplir une tâche plus importante. A la fin, votre VI principal contient une série de sous-VIs représentant les fonctions d'une application.

Chaque sous-VI pouvant être exécuté de façon autonome, indépendamment du reste de l'application, la mise au point de l'application en est simplifiée. En outre, plusieurs sous-VIs secondaires réalisant souvent les mêmes tâches communes à différentes applications, vous pouvez ainsi développer un ensemble spécialisé de sous-VIs correspondant exactement aux applications que vous serez amené à développer.

Les sections suivantes s'attachent de plus près à la face-avant, aux diagrammes, icônes, connecteurs et autres fonctions associées.

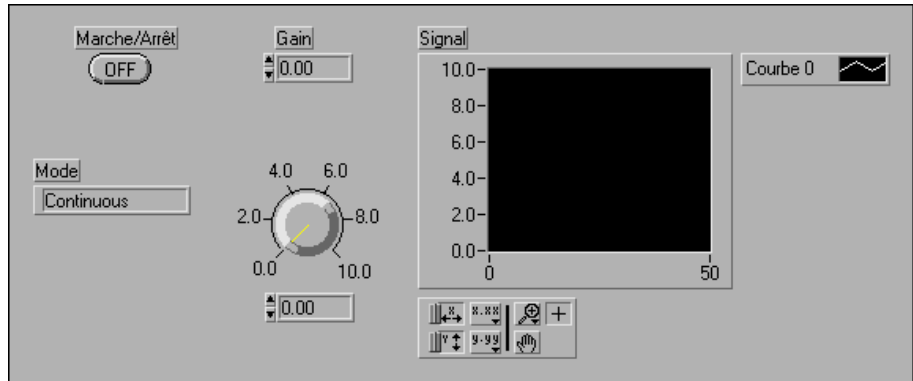


Remarque

Pour voir des exemples des différentes fonctionnalités utilisées par le G pour votre application LabVIEW ou BridgeVIEW, consultez le répertoire exemples.

Face-avant

L'interface utilisateur d'un VI est très semblable à celle d'un instrument réel : la face-avant. La face-avant d'un VI pourra par exemple ressembler à l'illustration suivante.



La face-avant d'un VI consiste en une combinaison de *commandes* et d'*indicateurs*. Les commandes simulent les périphériques d'entrée d'un instrument et fournissent des données au diagramme du VI. Les indicateurs simulent les périphériques de sortie affichant les données acquises ou générées par le diagramme du VI.

Vous pouvez ajouter des commandes et des indicateurs à la face-avant en les sélectionnant dans la palette flottante **Commandes** présentée dans l'illustration suivante.



Vous pouvez modifier la taille, la forme et la position des commandes et indicateurs. Chaque commande et chaque indicateur offrent un menu local permettant de modifier les divers attributs ou de sélectionner différentes commandes. Vous pouvez accéder à ce menu local en :

- **(Windows et UNIX)** double-cliquant sur l'objet avec la souris.
- **(Macintosh)** en cliquant sur la souris avec la touche <commande> enfoncée.

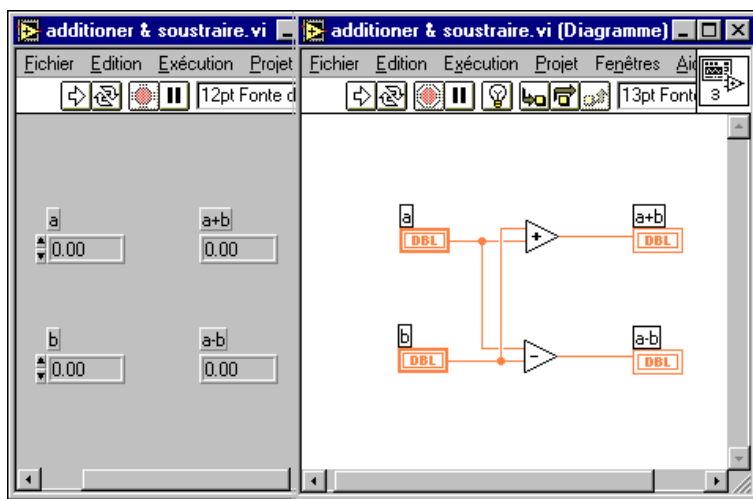
Pour plus d'informations sur la construction d'une face-avant, consultez le chapitre 2, *Edition des VIs* et la partie II, *Objets de la face-avant* de ce manuel.

Diagramme

Vous pouvez passer de la face-avant au diagramme en sélectionnant **Fenêtres»Diagramme**.

La fenêtre **Diagramme** renferme le diagramme du VI, c'est-à-dire le code source graphique d'un VI. Pour construire le diagramme, vous devez *relier* les objets qui transmettent ou reçoivent des données, remplissent certaines fonctions et contrôlent le flux de l'exécution.

Le VI suivant calcule la somme de deux chiffres et leur différence. Le diagramme présente plusieurs objets principaux de programmation d'un diagramme : les *nœuds*, les *terminaux* et les *fils de liaison*.



Lorsque vous placez une commande ou un indicateur sur la face-avant, un terminal correspondant apparaît sur le diagramme. Vous ne pouvez pas supprimer un terminal appartenant à une commande ou à un indicateur à partir du diagramme. Le terminal ne disparaît qu'avec la suppression de la commande ou de l'indicateur correspondant.

Les icônes des fonctions Ajouter et Soustraire disposent aussi de terminaux. Pensez aux terminaux comme étant des points d'entrée et de sortie. Les données que vous entrez dans les commandes (a et b) *sortent* de la face-avant via les terminaux de commande du diagramme. Les données *entrent* alors dans les fonctions Ajouter et Soustraire. Lorsque les fonctions d'ajout et de soustraction ont terminé leurs calculs internes, elles produisent de nouvelles valeurs sur leur terminal de *sortie*. Les données sont transmises aux terminaux des indicateurs et entrent à nouveau dans la face-avant, où elles sont affichées. Les données sortent d'un *terminal source* et entrent dans un *terminal destination*.

Les nœuds sont des éléments d'exécution de programme. Ils sont similaires aux instructions, opérateurs, fonctions et sous-programmes des langages de programmation conventionnels. Les fonctions Ajouter et Soustraire sont un exemple de nœud. Le G dispose d'une vaste bibliothèque de fonctions pour les calculs mathématiques, de comparaison, de conversion et d'entrée/sortie, pour n'en citer que quelques-unes.

Un autre type de nœud est la *structure*. Les structures sont des représentations graphiques des boucles et instructions conditionnelles des langages de programmation conventionnels. Le G comprend également des nœuds spéciaux servant à établir une liaison avec un code textuel externe et à comprendre des formules textuelles.

Les fils de liaison sont les chemins qu'empruntent les données pour aller des terminaux source vers les terminaux destination. Vous ne pouvez pas relier un terminal source à un autre terminal source, de même que vous ne pouvez pas relier un terminal de destination à un autre terminal de destination. Vous pouvez, par contre, relier une source à plusieurs destinations. Chaque fil de liaison a son propre style ou couleur, en fonction du type de données transmises. L'exemple précédent présente le style de fil de liaison correspondant à une valeur scalaire numérique : une ligne fine et unie.

Le principe gouvernant l'exécution d'un programme en G est appelé *flux de données*. Un nœud ne s'exécute que lorsque toutes les données nécessaires sont présentes sur les terminaux d'entrée ; le nœud ne fournit les données à tous ses terminaux de sortie qu'à la fin de son exécution ; les données passent immédiatement des terminaux source aux terminaux destination.

Le flux de données contraste avec la méthode de programmation séquentielle qui régit la programmation conventionnelle et où les instructions sont exécutées dans l'ordre de leur rédaction. Alors que l'exécution en programmation séquentielle est articulée autour des instructions, l'exécution de flux de données est *s'articule autour des données* ou est *asservie aux données*.

Pour plus d'informations sur l'utilisation d'objets de diagrammes servant à développer un programme, consultez la partie III, *Programmation par diagramme*.

Icône et connecteur

Lorsque l'icône d'un VI est placée dans le diagramme d'un autre VI, il devient alors un *sous-VI*, l'équivalent en G d'un sous-programme. Les commandes et indicateurs d'un sous-VI reçoivent les données du VI appelant et y retournent les données.

Le *connecteur* est un ensemble de terminaux correspondant aux commandes et indicateurs du sous-VI. L'*icône* est la représentation graphique d'un VI ou une description textuelle du VI ou de ses terminaux.

Le connecteur est semblable à la liste de paramètres d'un appel de fonction ; les terminaux du connecteur agissant comme des paramètres. Chaque terminal correspond à une commande ou à un indicateur particulier sur la face-avant. Un connecteur reçoit les données au niveau de ses terminaux d'entrée et les transmet au sous-VI via les commandes du sous-VI, ou reçoit les résultats au niveau de ses terminaux de sortie connectés aux indicateurs du sous-VI.

Chaque VI est représenté par une icône par défaut affichée dans l'angle supérieur droit des fenêtres de la face-avant et du diagramme. Comme dans l'illustration suivante.



Chaque VI dispose également d'un connecteur, auquel vous pouvez accéder en choisissant **Visualiser le connecteur** dans le menu local de l'icône au niveau de la face-avant. Lorsque vous affichez le connecteur pour la première fois, vous pouvez choisir un motif de connecteur proposé ou un autre de votre choix. Le connecteur est généralement équipé d'un terminal

pour chaque commande ou indicateur sur la face-avant. Vous pouvez attribuer jusqu'à 28 terminaux et laisser des terminaux sans connexion si vous prévoyez d'apporter des modifications au VI et d'y ajouter de nouvelles entrées ou sorties.

Pour plus d'informations, consultez le chapitre 3, *Utilisation des sous-VIs*.

Aide

La fenêtre d'aide contient des rubriques d'aide pour les fonctions, les constantes, les sous-VIs, les commandes et les indicateurs ainsi que pour les éléments de menus. Pour afficher cette fenêtre, choisissez **Visualiser l'aide** dans le menu **Aide** ou appuyez sur <Ctrl-h> (**Windows**), <commande-h> (**Macintosh**), <meta-h> (**Sun**) ou <Alt-h> (**HP-UX**). Si votre clavier est équipé d'une touche <Aide>, appuyez sur cette touche au lieu d'utiliser les combinaisons mentionnées ci-dessus. Placez votre curseur sur l'icône d'une fonction, d'un sous-VI ou d'un VI (y compris une icône de VI ouvert, présentée dans le coin supérieur droit de la fenêtre de VI) pour voir l'information d'aide.



La sélection de **Aide>Verrouiller l'aide** ou le fait de cliquer sur l'icône de verrouillage en bas de la fenêtre verrouille le contenu de la fenêtre d'aide. Lorsque vous verrouillez ce contenu, le fait de passer sur une autre fonction ou icône n'entraîne pas de changement dans la fenêtre d'aide. Sélectionnez **Verrouiller l'aide** ou cliquez à nouveau sur l'icône de verrouillage pour désactiver cette commande.

Aide pour la face-avant

Lorsque vous placez le curseur sur une commande ou un indicateur, la fenêtre d'aide en affiche la description. Il est toujours judicieux d'entrer des descriptions pour les commandes ou indicateurs au moment de la création d'un VI. Pour plus d'informations, consultez la section *Création de descriptions d'objets* du chapitre 2, *Edition des VIs*.

Si vous maintenez le curseur sur une icône de VI dans l'angle supérieur droit de la face-avant pendant quelques instants, la fenêtre d'aide affiche la rubrique d'aide correspondant à ce VI.

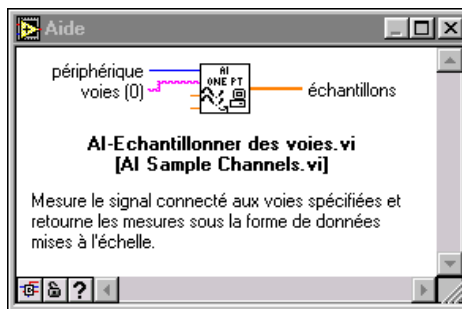
Aide pour le diagramme

Pour les fonctions et les sous-VIs, la fenêtre d'aide contient l'icône, les entrées et sorties, ainsi qu'une description de la fonctionnalité de ce nœud.

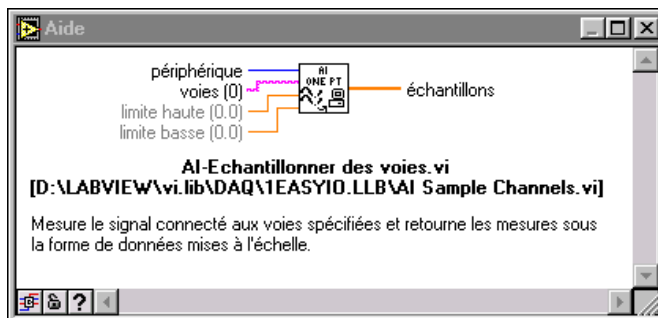


Vous pouvez afficher la fenêtre d'aide sous forme simplifiée ou détaillée en appuyant sur le second bouton en bas de la fenêtre d'aide ou en activant/désactivant la commande **Aide»Aide simple**.

La vue simplifiée met en valeur les connexions les plus importantes et donne moins de valeur aux connexions moins critiques. Dans ce type de vue, les étiquettes des connexions nécessaires en gras, alors que celles des connexions recommandées en texte normal. Les noms des connexions facultatives ne sont pas affichés et ne comportent qu'un petit onglet indiquant que certaines options existent au cas où l'utilisateur en aurait besoin. L'illustration suivante présente la vue simplifiée d'un VI d'acquisition de données.



Dans la vue détaillée, la fenêtre d'aide affiche toutes les entrées placées du côté gauche et les sorties placées du côté droit. Les étiquettes des entrées optionnelles sont en texte gris. L'illustration suivante présente la vue détaillée du même VI d'acquisition de données.



L'emplacement sur le disque du sous-VI est affiché sous l'icône en mode détaillé. En mode simplifié, seul le nom du sous-VI est affiché.

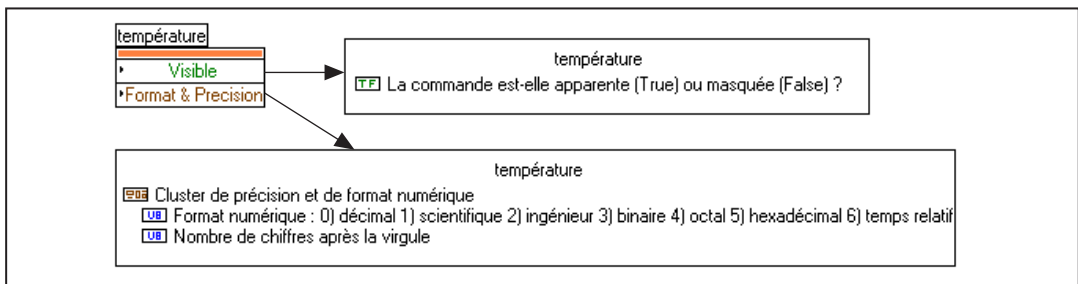
Si une entrée de fonction ne nécessite aucun câblage, la valeur par défaut est généralement indiquée entre parenthèses à côté du nom. Si la fonction peut accepter plusieurs types de données, la fenêtre d'aide affiche le type le plus courant.

Les noms de terminaux pour les nœuds de sous-VIs sont les étiquettes des commandes et indicateurs correspondants de la face-avant. Les valeurs par défaut des entrées n'apparaissent pas automatiquement dans la fenêtre d'aide. Il est toujours judicieux d'inclure la valeur par défaut entre parenthèses dans le nom des commandes lors de la création d'un sous-VI.

Lorsque vous placez l'outil Bobine sur un fil, la fenêtre d'aide affiche son type de données. Lorsque vous déplacez l'outil Bobine sur différentes zones de l'icône d'un VI, le terminal du connecteur correspondant est mis en évidence dans la fenêtre d'aide.

Aide pour les attributs

Si vous ne connaissez pas la signification d'un attribut particulier, vous pouvez consulter la fenêtre d'aide pour y trouver une description, son type de données et les valeurs qu'il accepte. Si l'attribut est d'un type plus compliqué (comme un cluster), la fenêtre d'aide affiche une description hiérarchique de la structure des données. L'illustration suivante présente un attribut node et l'aide correspondante, affichée lorsque vous placez le curseur sur différents noms d'attributs.



Référence en ligne



La fenêtre d'aide affiche les informations les plus importantes dans un format condensé. Pour accéder à des informations plus détaillées, choisissez **Aide»Référence en ligne**. Pour la plupart des objets du diagramme, vous pouvez sélectionner **Référence en ligne** dans le menu local de cet objet. Vous pouvez également accéder à cette information en cliquant sur le point d'interrogation (comme celui de gauche) situé en bas de la fenêtre d'aide.

Pour plus d'informations sur la création de fichiers d'aide en ligne personnalisés, consultez le chapitre 5, *Impression et documentation des VIs*.

Edition des VIs

Ce chapitre présente les principes de base pour la construction et l'utilisation de VIs et contient des informations sur les palettes et menus. Vous y trouverez également une description des tâches de base telles que la création d'objets, le changement d'outils, ainsi que l'ouverture, l'exécution et l'enregistrement des VIs.

L'environnement graphique

Les VIs sont construits à l'aide des palettes et des menus. Vous pouvez choisir des objets dans les palettes et les placer sur la face-avant ou dans le diagramme. Ces objets peuvent être déplacés ou modifiés à l'aide des outils ou des menus.

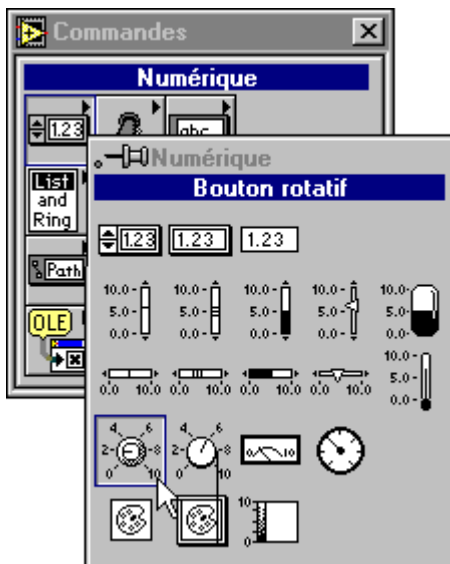


Remarque

Si vous utilisez BridgeVIEW, vous devez vous assurer que ces palettes sont bien modifiées en Basic G. Sélectionnez Editer»Sélectionner le type de palette. Choisissez ensuite la palette Basic G dans le menu.

Vous pouvez créer des objets sur la face-avant et dans le diagramme en les sélectionnant dans les palettes de **Commandes** et de **Fonctions**. Par exemple, pour créer un bouton rotatif sur une face-avant, sélectionnez-le

dans la palette **Numérique** de la palette **Commandes**, tel qu'indiqué dans l'illustration suivante.



Lorsque vous déplacez la flèche de sélection sur un objet de la palette, le nom de l'objet apparaît en haut de la palette. Dans l'illustration précédente, le bouton rotatif est sélectionné. A ce moment, si vous cliquez et placez la souris sur la face-avant tout en maintenant le bouton enfoncé, un bouton rotatif apparaît à l'endroit où vous relâchez la souris. Vous pouvez aussi cliquer et relâcher sur l'objet sur la palette. Lorsque vous passez au-dessus de la face-avant, vous pouvez voir la commande surlignée. Vous pouvez redimensionner un objet dès que vous le créez en cliquant et en le faisant glisser.



Remarque

Si vous avez besoin de plusieurs fonctions de la même palette, vous pouvez la laisser ouverte. Pour garder une palette ouverte, sélectionnez la punaise de l'angle supérieur gauche de la palette. Une fois la palette ouverte et verrouillée, une barre de titre apparaît, vous permettant de facilement déplacer la palette. Lorsque vous quittez votre environnement de développement graphique, l'emplacement des palettes est enregistré et conservé pour la prochaine fois que vous ouvrirez votre application.

*Si vous cliquez avec le bouton droit de la souris au niveau d'un endroit vide de la face-avant ou du diagramme pour ouvrir une palette locale, une copie temporaire des palettes **Commandes** et **Fonctions** apparaît. Si*

vous utilisez un moniteur de petite taille, vous pouvez fermer les palettes Commandes et Fonctions et créer des objets en n'utilisant que ces palettes locales.

Vous pouvez positionner la face-avant et le diagramme côte à côte ou l'un au-dessus de l'autre en sélectionnant la commande Mosaïque du menu Fenêtres.

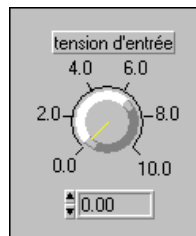
Lorsque vous créez des objets sur la face-avant, ces objets apparaissent avec une étiquette vierge prête à recevoir le nom du nouvel objet. Pour donner un nom à cet objet, il vous suffit de saisir ce nom. Une fois le nom entré, terminez la saisie en appuyant sur la touche <Entrée> du pavé numérique. Si vous ne disposez pas de pavé numérique, vous pouvez cliquer sur le bouton **Entrée** de la barre d'outils ou cliquer n'importe où à l'extérieur de l'étiquette.



Remarque

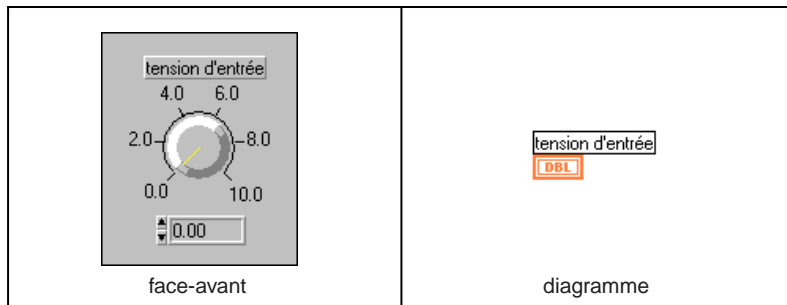
Si vous n'entrez pas de texte dans l'étiquette d'une commande lorsque vous la créez, l'étiquette disparaît dès que vous cliquez ailleurs. Vous pouvez afficher cette étiquette par la suite en cliquant à l'aide du bouton droit de la souris sur la commande pour ouvrir un menu local et en choisissant l'option Visualiser>Etiquette.

L'illustration suivante présente un exemple du résultat de la sélection de l'option **Visualiser>Etiquette** dans le menu local d'une commande.



Lorsque vous créez un objet sur la face-avant, un terminal correspondant est créé sur le diagramme. Ce terminal est utilisé pour lire les données d'une commande ou pour envoyer des données à un indicateur.

Si vous choisissez l'option **Fenêtres»Diagramme**, vous pouvez voir le diagramme correspondant à la face-avant. Ce diagramme contient les terminaux de toutes les commandes et de tous les indicateurs de la face-avant.



Palette d'outils

Un *outil* est un mode de fonctionnement spécial du curseur de la souris. Les outils sont utilisés pour définir l'opération qui va être réalisée lorsque l'utilisateur clique dans la face-avant ou le diagramme avec le bouton gauche de la souris.

Plusieurs outils de la palette flottante **Outils** sont présentés dans l'illustration suivante. Vous pouvez placer cette palette n'importe où, ou bien la fermer temporairement en cliquant sur la case de fermeture (la petite case contenant un "x" dans l'angle supérieur droit de la palette). Une fois la palette fermée, vous pouvez y accéder à nouveau en choisissant l'option **Fenêtres»Palette d'outils**. Vous pouvez également afficher une version temporaire de la palette **Outils** à l'emplacement du curseur en cliquant sur le bouton gauche de la souris tout en maintenant les touches <Ctrl-Maj> (**Windows**), <commande-Maj> (**Macintosh**), <meta-Maj> (**Sun**) ou <Alt-Maj> (**HP-UX**) enfoncées.



Les outils de la palette **Outils** et leurs fonctions/descriptions sont :



Outil Doigt : permet de modifier la valeur d'une commande ou de sélectionner le texte d'une commande.



Outil Flèche : permet de positionner, redimensionner et sélectionner les objets.



Outil Texte : permet de saisir le texte et de créer des étiquettes libres.



Outil Bobine : permet de connecter les objets dans le diagramme.



Outil Menu local : permet d'ouvrir le menu local pour un objet.



Outil Main : permet de faire défiler la fenêtre sans utiliser les barres de défilement.



Outil Point d'arrêt : permet de définir des points d'arrêt pour les VIs, fonctions, boucles, séquences et conditions.



Outil Sonde : permet de placer des sondes sur les fils.



Outil Pipette : permet de copier des couleurs à coller avec l'outil Pinceau.



Outil Pinceau : permet de définir les couleurs de premier plan et d'arrière-plan.

Pour changer d'outil en mode d'édition, utilisez une des méthodes suivantes :

- Cliquez sur l'outil souhaité de la palette **Outils**.
- Utilisez la touche <Tab> pour sélectionner successivement les outils les plus courants.
- Appuyez sur la barre d'outils pour basculer entre l'outil Doigt et l'outil Flèche lorsque la face-avant est active et entre l'outil Bobine et l'outil Flèche lorsque le diagramme est actif.

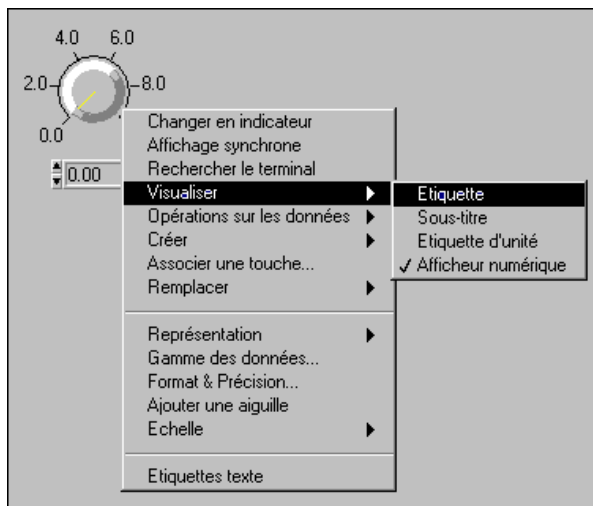
Utilisation des menus

La *barre de menus* en haut de la fenêtre de VI (ou en haut de l'écran Macintosh) contient plusieurs *menus déroulants*. Vous pouvez accéder aux menus déroulants d'une barre de menu en cliquant sur un des éléments de cette barre. Les menus déroulants sont normalement de nature générale et contiennent les commandes les plus courantes dans les applications informatiques, telles **Ouvrir**, **Enregistrer**, **Copier** ou **Coller** ainsi que d'autres commandes spécifiques à l'éditeur. Certains menus contiennent également des raccourcis clavier.

Le menu le plus souvent utilisé est le *menu local* d'objet. Presque chaque objet en G, de même que les espaces de face-avant et diagramme vides, est accompagné d'un menu local contenant différentes commandes. Il est préférable, dans la mesure du possible, de sélectionner une commande ou un menu dans un menu local d'objet.

Menus locaux

Comme vous avez pu le voir dans la section précédente, un menu local est associé à chaque objet en G. Un menu local est invoqué en cliquant sur l'objet avec le bouton droit de la souris (sur Macintosh, maintenez la touche <commande> enfoncée tout en cliquant sur l'objet avec la souris). Vous pouvez changer l'apparence et le comportement de l'objet grâce aux éléments du menu qui apparaît.



Edition des VIs

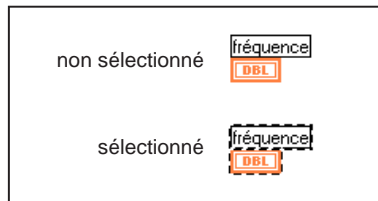
Après vous être familiarisé avec les techniques d'édition, consultez le chapitre 8, *Introduction aux objets de la face-avant* pour plus d'informations sur la construction des faces-avant et le chapitre 17, *Introduction au diagramme* pour plus d'informations sur la construction des diagrammes.

Sélection des objets

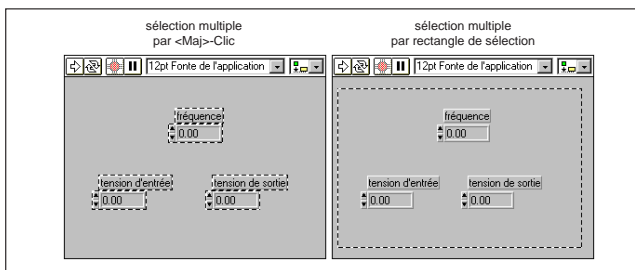


outil
Flèche

Pour la plupart des opérations d'édition, telles que le déplacement, la copie et la suppression, vous devez *sélectionner* un objet. Pour sélectionner un objet, utilisez l'outil Flèche et cliquez avec le bouton de la souris lorsque la pointe de la flèche se trouve sur l'objet. Une fois l'objet sélectionné, il est entouré d'une ligne pointillée mobile appelée *marquise*.



Pour sélectionner plusieurs objets, cliquez sur chaque objet supplémentaire en maintenant la touche <Maj> enfoncée. Vous pouvez également désélectionner un objet sélectionné en cliquant dessus tout en maintenant la touche <Maj> enfoncée. Une autre façon de sélectionner un ou plusieurs objets est de faire glisser un rectangle de sélection autour du/des objets en question, comme indiqué dans l'illustration suivante.



Les objets vides doivent généralement être entièrement englobés dans le rectangle pour être sélectionnés. En général, un objet est sélectionné si une partie est en contact avec le rectangle de sélection. En maintenant la touche <Maj> enfoncée tout en faisant glisser le rectangle de sélection, vous

pouvez sélectionner des objets supplémentaires ou désélectionner un objet englobé dans le rectangle.

Le fait de cliquer sur un objet non sélectionné ou dans une zone vide annule la sélection en cours. Vous ne pouvez pas sélectionner un objet de la face-avant et un objet du diagramme en même temps. Cependant, vous pouvez sélectionner plusieurs objets d'une même face-avant ou d'un diagramme.

Glisser-déposer des VIs, des images et du texte

Vous pouvez faire glisser des VIs depuis le système de fichiers vers un diagramme pour créer un appel de sous-VIs à ce VI. Cette fonction est également utilisable pour les commandes personnalisées, les définitions de type et variables globales. Vous pouvez également faire glisser du texte et des images depuis d'autres applications pour les copier dans une face-avant ou un diagramme.

Sous Windows 3.1, le support du glisser est limité au passage de VIs et commandes depuis le gestionnaire de fichiers. Sous Windows 95/NT, supportant le modèle OLE 32 bits (*Object Linking and Embedding*), vous pouvez faire glisser du texte et des images depuis des applications supportant le modèle OLE, comme vous pouvez le faire depuis le Gestionnaire de fichiers/Explorateur.

Sur Macintosh, votre système doit contenir le Gestionnaire de glisser-déposer pour que cette fonction soit utilisable. Ce gestionnaire est intégré au Système 7.5 et aux suivants (il est également disponible pour les Systèmes 7.0 à 7.5 sous la forme d'une extension disponible auprès d'Apple).

Vous pouvez utiliser les fonctions glisser-déposer suivantes au sein d'un même programme :

- Faire glisser des commandes et indicateurs depuis une face-avant vers un diagramme pour créer des constantes (et vice versa).
- Faire glisser un sous-VI vers une constante ou commande de chemin pour placer le chemin complet du VI dans la commande ou constante.

Vous pouvez utiliser les fonctions glisser-déposer suivantes à partir d'autres applications :

- Faire glisser un fichier dans une constante ou commande de chemin pour placer son chemin complet.
- Faire glisser un fichier d'enregistrement de données dans un Refnum de fichiers d'enregistrement de données pour créer un cluster contenant la structure de données dans le fichier d'enregistrement de données.

- Faire glisser un fichier graphique dans un menu déroulant d'image, une face-avant ou un diagramme pour y placer l'image qu'il contient.
- Faire glisser un fichier VI dans le diagramme d'un VI pour le déposer comme sous-VI.
- Faire glisser un fichier de commande personnalisé dans la face-avant pour y placer la commande personnalisée contenue dans ce fichier.
- **(Windows 95/NT)** Faire glisser du texte depuis une source OLE pour le placer dans une commande ou constante type chaîne de caractères, face-avant, diagramme ou étiquette.
- **(Windows 95/NT)** Faire glisser une image depuis une source OLE pour la placer dans un menu déroulant d'image, une face-avant ou un diagramme.
- **(Macintosh)** Faire glisser une coupure de texte ou du texte sélectionné dans n'importe quelle application pour la placer dans une commande ou constante type chaîne de caractères, une face-avant, un diagramme ou une étiquette.
- **(Macintosh)** Faire glisser une coupure d'image ou une image sélectionnée dans une autre application pour la placer dans un menu déroulant d'image, une face-avant ou un diagramme.



Remarque

Si vous placez un objet avec succès, la destination est mise en surbrillance.

Pour plus d'informations sur les fonctions de glisser-déposer, consultez le mode d'emploi de votre système Windows ou Macintosh.

Positionnement d'objets



Vous pouvez positionner un objet en cliquant dessus avec l'outil **Flèche** et en le faisant glisser vers l'emplacement souhaité.

Si vous maintenez la touche <Maj> enfoncée et faites glisser un objet, la direction du mouvement est restreinte horizontalement ou verticalement, en fonction de la première direction suivie par l'objet.

Vous pouvez déplacer des objets sélectionnés par incréments réduits et précis en appuyant sur la touche de direction appropriée du clavier une fois pour un mouvement d'un pixel. Maintenez la touche de direction enfoncée pour répéter l'action. Maintenez la touche <Maj> enfoncée en même temps que la touche de direction pour déplacer l'objet plus rapidement.

Si vous changez d'avis sur le sens du déplacement alors que vous êtes en train de déplacer l'objet, continuez à le faire glisser jusqu'à ce que le

curseur se trouve en dehors de toutes les fenêtres ouvertes et que la ligne pointillée disparaisse, puis relâchez le bouton de la souris. Cela a pour effet d'annuler l'opération, l'objet restant à sa position d'origine. Si votre écran est rempli, la barre de menus est un endroit pratique et sûr pour relâcher le bouton de la souris et annuler une opération.



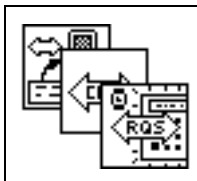
Remarque

L'outil Flèche permet également d'agrandir les objets. Faites donc attention lorsque vous cliquez avec la souris sur un objet. Déplacez un objet en sélectionnant son centre, agrandissez-le à partir des angles.

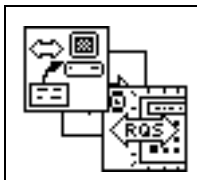
Commande Réorganiser



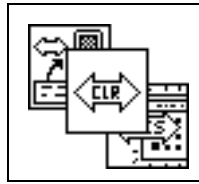
Vous pouvez également placer des objets au-dessus d'autres objets. Plusieurs commandes peuvent être utilisées pour déplacer des objets les uns par rapport aux autres dans **Réorganiser**, à l'extrémité droite de la barre d'outils en haut de la face-avant. Par exemple, si vous avez trois objets empilés les uns sur les autres : l'objet 1 est en bas de la pile et l'objet 3 est en haut.



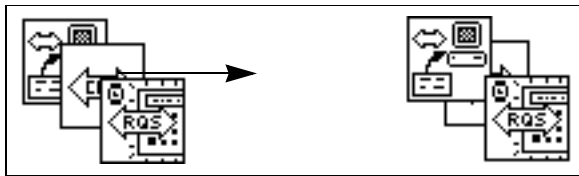
Placer en premier plan déplace l'objet sélectionné de la pile vers le haut. Si l'objet 1 est sélectionné, l'objet 1 monte en haut de la pile, avec l'objet 3 dessous et l'objet 2 tout en bas.



Si l'objet 2 est sélectionné ensuite, **Placer en premier plan** place l'objet 2 en haut, avec l'objet 1 dessous et l'objet 3 tout en bas.



Placer devant déplace l'objet sélectionné d'une place vers le haut dans la pile. Si l'on considère que l'objet 1 est en bas de la pile et l'objet 3 est en haut, la sélection de cette commande pour l'objet 1 place l'objet 2 en bas, l'objet 3 en haut et l'objet 1 au milieu.

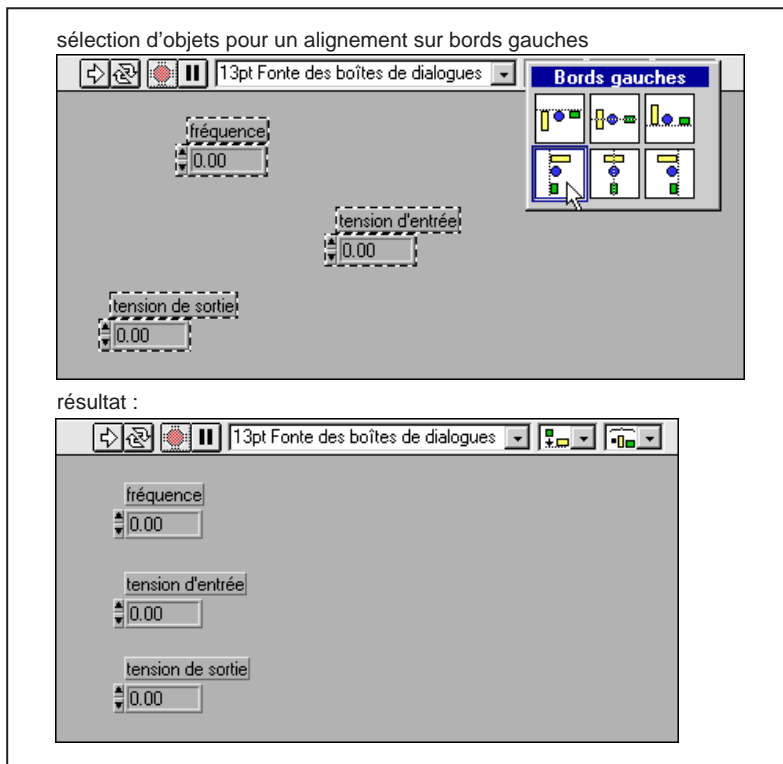


Placer en arrière-plan et **Placer derrière** fonctionnent d'une façon semblable à **Placer en premier plan** et **Placer devant** mais déplacent les éléments vers le bas de la pile.

Alignement d'objets



Sélectionnez les objets à aligner et choisissez l'axe sur lequel vous souhaitez les aligner dans le menu déroulant **Aligner les objets**, tel qu'illustré ci-contre. L'illustration suivante montre comment sélectionner des objets à aligner et le résultat de la sélection.



Vous pouvez aligner un objet sur l'axe vertical en fonction des bords gauches, droits ou du centre des objets. Vous pouvez également aligner un objet sur l'axe horizontal en fonction des bords supérieurs, inférieurs ou du centre des objets. Votre sélection devient l'option d'alignement courante, indiquée par une bordure sombre autour de l'élément dans la palette. Sur Macintosh, <commande-A> répète la sélection d'alignement sans avoir à utiliser le menu.

Répartition des objets



Pour répartir ou espacer des objets, sélectionnez les objets à répartir et choisissez la manière de les répartir dans le menu déroulant **Répartir les objets**. Vous pouvez, en plus de la possibilité de répartir des objets, en rendant leurs bords ou centres équidistants, ajouter ou supprimer des espaces horizontalement ou verticalement entre les objets grâce à quatre éléments placés sur la droite du menu déroulant. Sur Macintosh, <commande-D> répète la répartition sur n'importe quelle sélection.

Copie des objets

Ils existe trois méthodes de base pour copier un objet : le copier-coller, le clonage et le glisser-déposer.

Pour copier et coller dans un VI ou entre VIs, sélectionnez l'objet avec l'outil Flèche et choisissez **Editer»Couper** ou **Editer»Copier**. Cliquez ensuite sur la zone dans laquelle placer la copie et choisissez l'option **Editer»Coller**. Vous pouvez copier plusieurs objets en même temps en plaçant un cadre de sélection autour des objets avant de les copier. Vous pouvez encore copier du texte et des images à partir d'autres applications et les coller dans une face-avant ou un diagramme.

Pour cloner un objet, cliquez sur l'objet avec l'outil Flèche, tout en maintenant la touche <Ctrl> (**Windows**), <option> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) enfoncée et faites glisser la copie vers son nouvel emplacement. Sous UNIX, vous pouvez également cloner des objets en cliquant sur le bouton du milieu et en faisant glisser la souris sans avoir à utiliser de touches de modification.

Lorsque vous clonez ou copiez des objets ayant des étiquettes, le logiciel donne aux copies le même nom que l'original et y ajoute le mot *copie* (ou *copie 1*, *copie 2*, etc. pour des copies de copies).

Si vous copiez une commande de face-avant, un nouveau terminal est créé dans le diagramme. Remarquez que les variables locales et les attributs nodes de la commande d'origine ne sont pas copiés. Lorsque vous utilisez l'option **Editer»Copier** ou **Editer»Coller** sur une variable locale, l'objet de la face-avant est également copié. Si vous clonez une variable locale, une nouvelle référence vers la commande d'origine est créée. Pour plus d'informations, consultez le chapitre 22, *Attribute Nodes* et le chapitre 23 *Variables globales et locales* de ce manuel.

Vous pouvez cloner, copier ou faire glisser et déposer des commandes de la face-avant vers n'importe quel diagramme pour y créer une constante du même type. Vous pouvez, de la même façon, faire glisser des constantes personnalisées d'un diagramme vers une face-avant pour créer des commandes.

Suppression d'objets

Pour supprimer un objet, sélectionnez l'objet et choisissez l'option **Editer»Effacer** ou appuyez sur la touche <Retour arrière> ou la touche <Supprimer>. Vous ne pouvez supprimer des terminaux de diagramme qu'en supprimant les commandes et indicateurs correspondants dans la face-avant.

Bien que vous puissiez supprimer la plupart des objets, vous ne pouvez pas supprimer des parties d'une commande ou d'un indicateur telles que les étiquettes ou afficheurs numériques. Vous pouvez, cependant, masquer ces composants en désélectionnant l'option **Visualiser»Étiquette** ou **Visualiser»Afficheur numérique** dans le menu local de l'objet.

Si vous supprimez une structure, telle qu'une boucle **While**, celle-ci est supprimée avec son contenu. Pour ne supprimer que la structure tout en conservant le contenu, cliquez sur le bord de la structure et choisissez **Supprimer la boucle While** (ou un autre nom de structure) dans le menu local. Cette action entraîne la suppression de la structure mais pas de son contenu et reconnecte automatiquement les fils traversant la bordure de la structure.

Nom des objets

Les étiquettes sont des pavés de texte nommant les composantes de la face-avant et du diagramme. Il existe deux types d'étiquettes : les étiquettes *asservies* et les étiquettes *libres*. Les étiquettes asservies appartiennent à un objet particulier, l'accompagnent dans ses déplacements et n'annotent que cet objet. Vous pouvez masquer ces étiquettes mais ne pouvez pas les copier ou les supprimer indépendamment de leur propriétaire. Les étiquettes libres ne sont pas, elles, liées à un objet particulier et peuvent être créées, déplacées ou supprimées librement. Utilisez ces étiquettes pour commenter votre face-avant et votre diagramme. Vous pouvez utiliser l'outil Texte, au curseur en forme de I majuscule, pour créer des étiquettes libres ou pour modifier n'importe quel type d'étiquette.



Sous-titres de faces-avant

Un nom est associé à chaque objet de la face-avant. Ce nom peut permettre de distinguer cet objet des autres. Ce nom est également utilisé sur le terminal, les variables locales et les attribut nodes de l'objet.

Si l'objet est une commande, le texte de l'étiquette désigne le fil de liaison auquel la commande est reliée sur le diagramme. Si l'objet est un indicateur et est connecté au cadre connecteur, l'étiquette désigne les fils de liaison

connectés au terminal du cadre connecteur sur les diagrammes des VIs appelants de ce VI. Cela signifie que, lorsque le texte de l'étiquette change, LabVIEW doit recompiler le VI et éventuellement recompiler les VIs qui appellent ce VI.

Les objets de face-avant peuvent également disposer de sous-titres. Le sous-titre n'affecte pas le nom de l'objet et peut être utilisé comme nom décrivant mieux l'objet. Vous pouvez également afficher, masquer ou modifier le sous-titre par programmation grâce aux attributs nodes. Les modifications apportées au sous-titre n'entraînent pas la recompilation du VI ou de ses appelants.

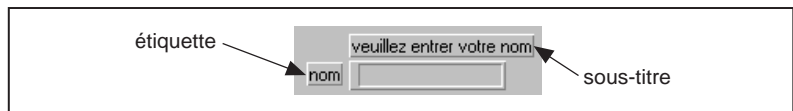


Figure 2-1. Boîte de dialogue Etiquette de la face-avant

L'étiquette est affichée lorsque vous déposez un objet de face-avant. Pour accéder au sous-titre pendant que vous modifiez l'objet, cliquez en utilisant le bouton droit de la souris et choisissez l'option **Visualiser»Sous-titre** dans le menu local, tel que présenté dans l'illustration suivante.

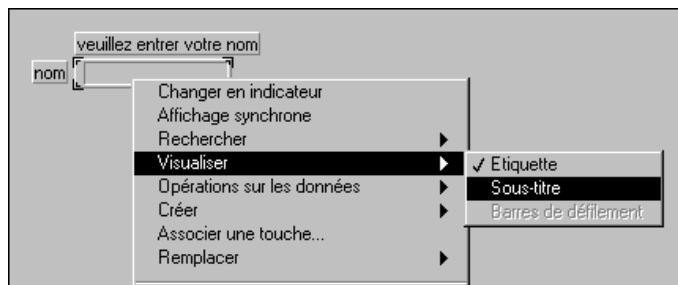


Figure 2-2. Menu local d'une étiquette

Pour afficher, masquer ou modifier le sous-titre au moyen d'un programme, utilisez les attributs indiqués dans l'illustration ci-dessous.



Figure 2-3. Objet sur le diagramme, avec le sous-titre visible

Lorsque vous affichez la fenêtre d'aide pour un objet (en choisissant **Aide»Visualiser l'aide**), son sous-titre apparaît dans une fenêtre et son étiquette apparaît entre crochets, comme présenté dans l'illustration suivante.

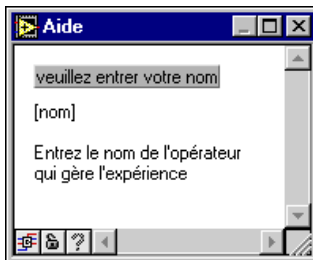
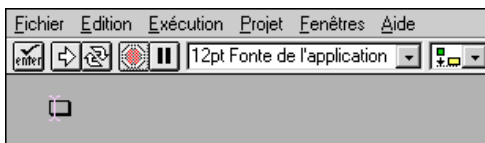


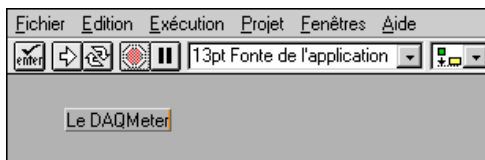
Figure 2-4. Boîte de dialogue d'aide contextuelle

Étiquettes libres

Pour créer une étiquette libre, sélectionnez l'outil Texte dans la palette **Outils** et cliquez dans un endroit vierge.



Une petite boîte apparaît alors avec un curseur dans la marge gauche, prêt à accepter la saisie. Entrez le texte devant apparaître dans l'étiquette. Utilisez le bouton **Entrée** de la barre d'outils ou du clavier pour terminer l'opération d'édition.



Vous pouvez également mettre fin à la saisie en cliquant n'importe où à l'extérieur de l'étiquette. Si vous ne tapez pas de texte dans l'étiquette, celle-ci disparaît dès que vous cliquez à un autre endroit.

**Remarque**

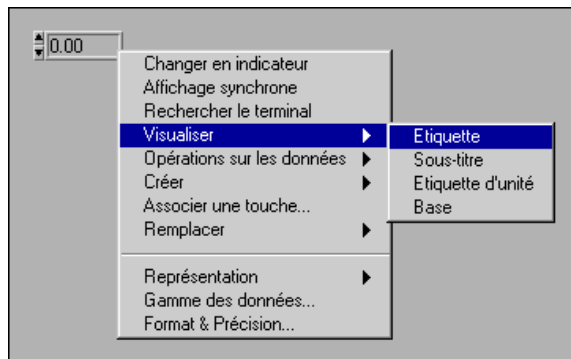
Si vous placez une étiquette ou un autre objet sur (recouvrant partiellement) une commande ou un indicateur, le rafraîchissement de l'écran est ralenti et un scintillement de la commande ou de l'indicateur peut avoir lieu. Pour éviter ce problème, ne superposez pas un objet de face-avant sur une étiquette ou un autre objet.

Vous pouvez copier le texte d'une étiquette en le sélectionnant à l'aide de l'outil Texte. Double-cliquez sur le texte avec l'outil Texte pour sélectionner par mot(s). Triple-cliquez sur le texte avec l'outil Texte pour sélectionner l'étiquette entière. Une fois le texte sélectionné, choisissez **Editer»Copier** pour copier le texte dans le Presse-papiers. Vous pouvez maintenant sélectionner le texte d'une seconde étiquette et choisir **Editer»Coller** pour remplacer le texte en surbrillance de la seconde étiquette par le texte du Presse-papiers. Pour créer une nouvelle étiquette avec le texte du Presse-papiers, cliquez sur l'écran avec l'outil Texte à l'endroit auquel vous voulez placer la nouvelle étiquette et choisissez **Editer»Coller**.

Lorsque vous créez une commande ou un indicateur sur la face-avant, une étiquette asservie vierge l'accompagne, vous permettant d'entrer le nom de la nouvelle commande ou du nouvel indicateur. Si vous ne tapez pas de texte dans l'étiquette, celle-ci disparaît dès que vous cliquez à un autre endroit. Vous pouvez à nouveau afficher l'étiquette en cliquant avec la souris sur la commande ou l'indicateur et en choisissant **Visualiser»Etiquette** dans le menu local.

Les structures et les fonctions sont accompagnées d'une étiquette masquée par défaut.

Pour afficher une étiquette masquée, cliquez avec la souris sur l'objet et choisissez **Visualiser»Etiquette** dans le menu local, tel que présenté dans l'illustration suivante.



L'étiquette apparaît, attendant la saisie du texte. Si vous ne tapez pas de texte dans l'étiquette, celle-ci disparaît dès que vous cliquez à un autre endroit.

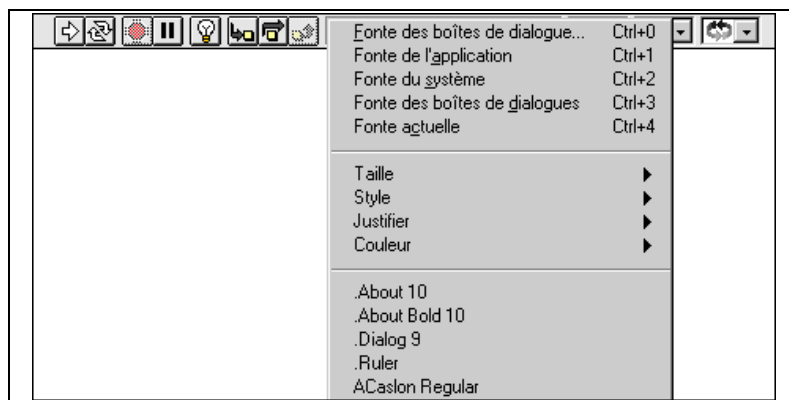


Remarque

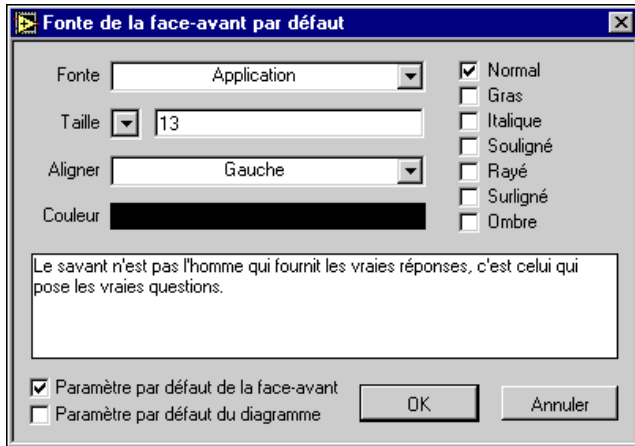
Sur le diagramme, les sous-VIs ne disposent pas d'étiquettes pouvant être modifiées. L'étiquette d'un sous-VI contient toujours le nom de ce sous-VI. Les étiquettes de fonctions, au contraire, peuvent être modifiées pour refléter l'utilisation faite de la fonction dans le diagramme. Par exemple, vous pouvez utiliser l'étiquette d'une fonction Ajouter pour documenter les quantités ajoutées ou la raison de leur ajout à cet endroit du diagramme. Pour plus d'informations, consultez la section Création de descriptions pour les VIs de ce chapitre.

Caractéristiques du texte

Vous pouvez modifier les attributs du texte à l'aide des commandes du menu **Fonte** de la barre d'outils, tel que présenté dans l'illustration suivante. Si vous sélectionnez des objets ou du texte et choisissez une commande de ce menu, les changements s'appliquent à toute la sélection. Si rien n'est sélectionné, les changements s'appliquent à la police par défaut, les nouvelles étiquettes utilisant alors la nouvelle police par défaut. La modification de la police par défaut n'entraîne pas le changement de la police des étiquettes existantes mais affectera les étiquettes créées par la suite.



Si vous sélectionnez **Fonte de dialogue...** dans le menu des fontes alors qu'une face-avant est active, la boîte de dialogue présentée dans l'illustration suivante apparaît. Si c'est un diagramme qui est actif au lieu d'une face-avant, l'option **Paramètre par défaut du diagramme** remplacera l'option **Paramètre par défaut de la face-avant** en bas de la boîte de dialogue.



Si la case **Paramètre par défaut de la face-avant** ou **Paramètre par défaut du diagramme** est cochée, les autres sélections effectuées dans cette boîte de dialogue sont utilisées pour les nouvelles étiquettes qui seront créées sur la face-avant ou le diagramme.

Dans l'illustration précédente, le mot **Application** apparaît dans le menu **Fonte**. Ce menu contient également les commandes **Fonte système**, **Fonte des boîtes de dialogue** et **Fonte actuelle**. La dernière commande du menu, **Fonte actuelle**, se rapporte au dernier style de police sélectionné.

Les fontes Application, Système et Boîte de dialogue sont utilisées pour des parties spécifiques de l'interface. Ces fontes sont prédéfinies et permettent de passer d'une plate-forme à une autre sans problème. Lorsque vous transférez des VIs contenant une de ces fontes sur une autre plate-forme, la fonte correspond autant que possible à celle de ce système.

Les fontes prédéfinies sont :

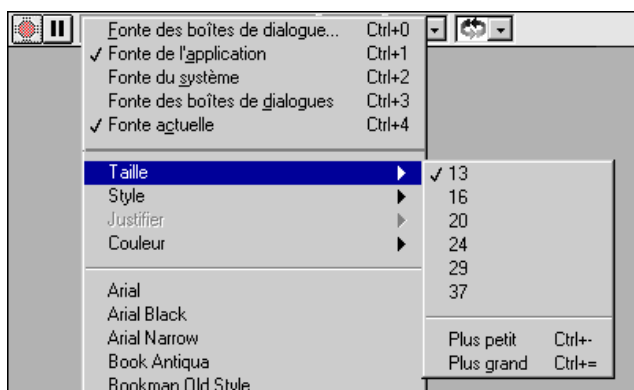
- Application : généralement utilisée comme police par *défaut*. Cette police est utilisée pour la palette **Commandes**, la palette **Fonctions** et dans le texte des nouvelles commandes.
- Système : utilisée pour les menus.
- Boîte de dialogue : utilisée pour le texte des boîtes de dialogue.

Si vous cochez les cases **Paramètre par défaut de la face-avant** et **Paramètre par défaut du diagramme**, la fonte sélectionnée devient la fonte utilisée pour la face-avant, le diagramme ou les deux (pour les nouvelles étiquettes). Vous pouvez utiliser ces cases pour définir des fontes différentes pour la face-avant et le diagramme. Par exemple, vous pouvez

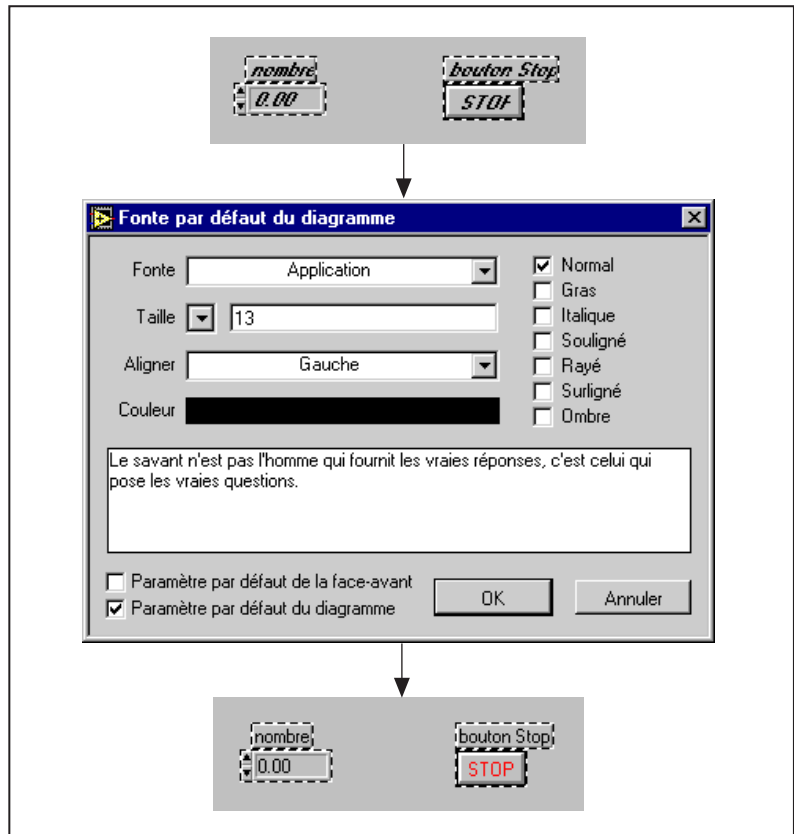
avoir une fonte de petite taille sur le diagramme et une fonte de grande taille sur la face-avant.

Pour plus d'informations sur le passage entre plates-formes pour ces trois types de fontes, consultez la section *Différences de résolution et de fonte* dans le chapitre 29, *Portabilité et localisation*. Pour plus d'informations sur le changement de la valeur par défaut pour ces trois types de fontes, consultez la section *Préférences de fonte* dans le chapitre 7, *Personnalisation de votre environnement*.

Le menu **Fonte** contient également les commandes **Taille**, **Style**, **Justifier** et **Couleur**, tel que présenté dans l'illustration suivante.



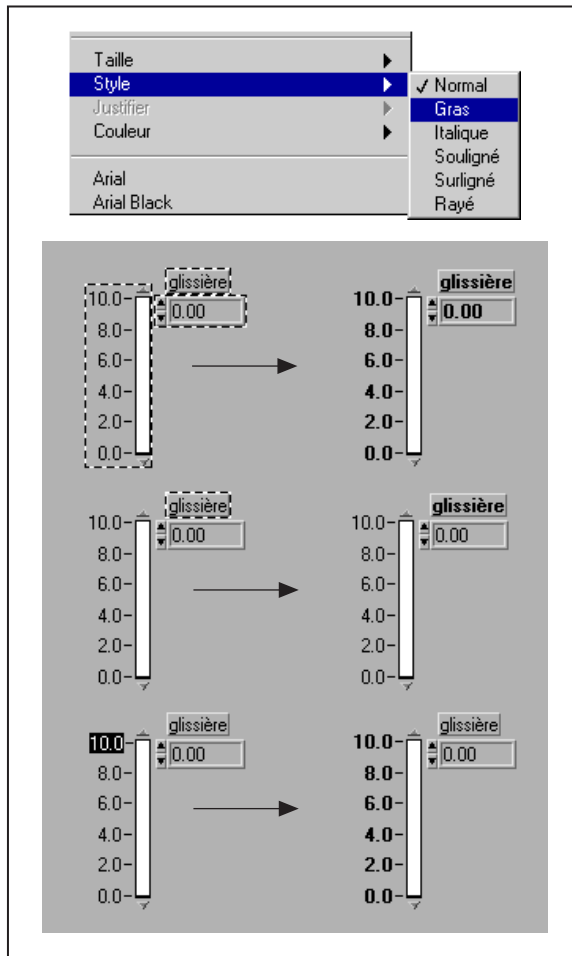
Les sélections effectuées dans ces sous-menus s'appliquent aux objets sélectionnés. Par exemple, si vous choisissez une nouvelle fonte alors qu'un bouton rotatif et un graphique sont sélectionnés, les étiquettes, échelles et afficheurs numériques adoptent tous la nouvelle fonte. L'illustration suivante présente une commande numérique et une commande booléenne passant de la fonte courante à la fonte Système.



Le G préserve autant d'attributs de fontes que possible lors de changements. Si vous attribuez la police Courier à plusieurs objets, ces objets gardent leur taille et leur style (si possible). De la même façon, le fait de changer la taille de textes utilisant des fontes différentes n'entraîne pas l'utilisation de la même fonte. Ces règles ne s'appliquent pas si vous sélectionnez une des fontes prédéfinies, la fonte courante ou lorsque vous utilisez la boîte de dialogue **Fonte**, qui modifiera uniquement la fonte des objets sélectionnés et les attributs l'accompagnant.

Lorsque vous travaillez avec des objets tels que des glissières, qui ont plusieurs blocs de texte, rappelez-vous que les sélections de texte affectent les objets ou le texte sélectionné. Par exemple, si vous sélectionnez la glissière toute entière en sélectionnant **Gras**, l'échelle, l'afficheur numérique et l'étiquette adoptent tous le style gras. Si vous ne sélectionnez que l'étiquette et choisissez **Gras**, seule l'étiquette adopte le style gras.

Si vous sélectionnez du texte dans un repère d'échelle, en choisissant **Gras**, tous les repères adoptent le style gras. Ces trois types de changement sont présentés dans l'illustration suivante.



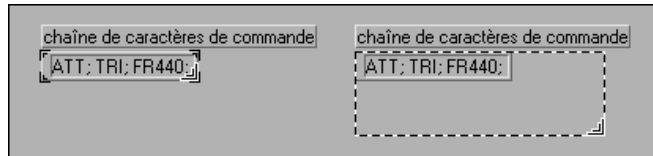
Redimensionnement des objets

Vous pouvez changer la taille de la plupart des objets. Lorsque vous placez l'outil Flèche sur un objet redimensionnable, les triangles de redimensionnement apparaissent aux angles des objets rectangulaires alors que des cercles de redimensionnement apparaissent pour des objets circulaires, tel que présenté dans l'illustration suivante.



Pour redimensionner un objet circulaire, tirez sur les cercles à l'aide du curseur. Pour redimensionner un objet rectangulaire, tirez sur les triangles de redimensionnement à l'aide du curseur.

Lorsque vous placez l'outil au-dessus d'une zone de redimensionnement, l'outil Flèche se transforme en *curseur de redimensionnement*. Cliquez et faites glisser le curseur jusqu'à ce que la bordure pointillée corresponde à la taille souhaitée, tel que présenté dans l'illustration suivante.



Pour annuler une opération de redimensionnement, continuez à faire glisser le cadre jusqu'à ce que le curseur se trouve en dehors de la fenêtre et que le cadre pointillé disparaisse, puis relâchez le bouton de la souris. L'objet conserve alors sa taille d'origine.

Certains objets ne changent de taille qu'horizontalement ou verticalement ou conservent les mêmes proportions au moment du redimensionnement (comme les boutons rotatifs). Le curseur de redimensionnement apparaît de la même façon mais la ligne pointillée ne bouge que dans une direction. Pour restreindre la direction du changement horizontalement ou verticalement ou pour conserver les proportions de l'objet, maintenez la touche <Ma j> enfoncée et faites glisser l'objet.

Étiquettes

Les étiquettes peuvent être redimensionnées de la même façon que les autres objets, à l'aide du curseur de redimensionnement. Les étiquettes sont généralement *automatiquement redimensionnées*, le cadre de l'étiquette s'adaptant à la taille du texte. Le texte des étiquettes reste sur une ligne à moins que vous n'entriez un retour de chariot ou redimensionniez le pavé de texte. Choisissez **Ajuster à la taille du texte** dans le menu local de l'étiquette pour activer le redimensionnement automatique.

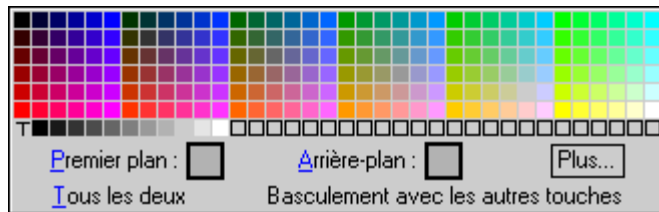
Ajout d'espace de travail

Pour insérer plus d'espace de travail dans votre face-avant ou diagramme, effectuez un <Ctrl-clic> (**Windows**), <option-clic> (**Macintosh**), <meta-clic> (**Sun**), <Alt-clic> (**HP-UX**) et définissez une zone avec l'outil Flèche. Sous UNIX, vous pouvez utiliser le bouton du milieu de la souris et insérer plus d'espace de travail en faisant glisser la souris sans avoir à utiliser de touches de modification. Après avoir cliqué, un rectangle marqué d'une ligne pointillée apparaît, définissant le nouvel espace de travail.

Coloriage des objets

Votre environnement de développement en G apparaît en noir et blanc, avec des nuances de gris ou en couleur, en fonction de votre moniteur. Vous pouvez changer la couleur de plusieurs objets graphiques (mais pas tous). Par exemple, les terminaux de diagrammes ainsi que les objets et fils de liaison de face-avant utilisent des codes de couleur permettant d'identifier le type de données transportées et ne peuvent donc pas être modifiés. Vous ne pouvez pas changer les couleurs en mode noir et blanc.

Pour changer la couleur d'un objet ou de l'arrière-plan d'une fenêtre, cliquez dessus avec la souris à l'aide de l'outil Pinceau de la palette **Outils**, tel qu'illustré ci-contre. La palette suivante apparaît en couleur.



Si vous déplacez le curseur dans la palette tout en appuyant sur le bouton de la souris, l'objet ou l'arrière-plan dont vous voulez changer la couleur prend la couleur pointée par le curseur, ce qui vous permet d'avoir un aperçu de l'objet dans sa nouvelle couleur. Si vous relâchez le bouton de la souris sur une couleur, l'objet conservera cette couleur. Pour annuler une modification de couleur, placez le curseur à l'extérieur de la palette avant de relâcher le bouton de la souris.



Si vous sélectionnez la case contenant un T, l'objet devient transparent, cette fonction permettant de placer des objets sur plusieurs couches. Par exemple, vous pouvez placer des commandes invisibles au-dessus d'indicateurs ou créer des commandes numériques sans l'apparence

tridimensionnelle standard. La transparence n'affecte que l'apparence d'un objet, celui-ci répondant normalement à la souris et aux touches.

Certains objets peuvent disposer d'un premier plan et un arrière-plan dont les couleurs peuvent être changées séparément. La couleur de premier plan d'un bouton rotatif, par exemple, est celle de la zone principale du cadran et la couleur d'arrière-plan celle du bord en biseau. L'affichage en bas de la boîte de sélection des couleurs indique si vous êtes en train de changer la couleur du premier-plan, de l'arrière-plan ou des deux. Une bordure noire autour de la couleur indique ce qui est sélectionné. Avec les valeurs par défaut, le premier plan et l'arrière-plan sont tous deux sélectionnés.

Pour alterner entre le premier plan et l'arrière-plan, vous pouvez appuyer sur <f> pour le *premier plan* et sur pour l'*arrière-plan*, ou sur <a> pour *les deux*. Le fait d'appuyer sur une autre touche fait également passer du premier plan à l'arrière-plan et un inversement.

Le choix de **Plus...** dans la palette **Couleur** entraîne l'apparition d'une boîte de dialogue permettant de personnaliser les couleurs. La boîte de dialogue **Davantage de couleurs** est présentée dans l'illustration suivante.



Chacune des trois composantes de couleurs que sont le rouge, le vert et le bleu, représente 8 bits d'une couleur codée sur 24 bits. Chaque composante a donc une fourchette allant de 0 à 255. Pour changer la valeur d'une composante de la couleur, vous pouvez double-cliquer sur cette couleur et entrer la nouvelle valeur. Pour changer une des couleurs de base, cliquez sur le rectangle de couleur et entrez votre choix. Les valeurs de composante pour la couleur sélectionnée apparaissent dans chaque affichage.

La dernière couleur sélectionnée dans la palette devient la couleur courante. Le fait de cliquer sur un objet avec l'outil Pinceau donne la couleur courante à cet objet.

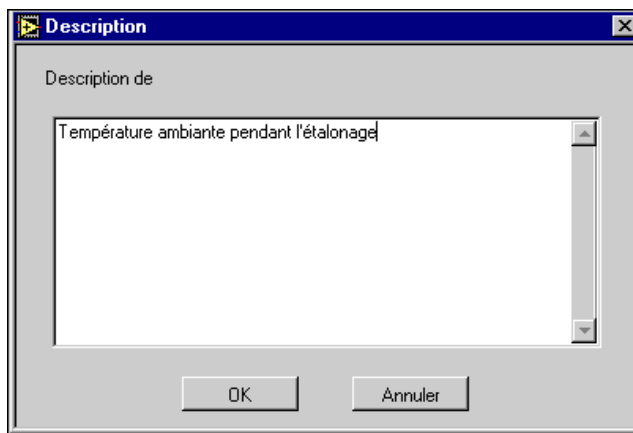
Vous pouvez également copier la couleur d'un objet et la donner à un second objet sans utiliser la palette **Couleur**. Pour ce faire, cliquez avec l'outil Pipette <Ctrl-clic> (**Windows**), <commande-clic> (**Macintosh**), <meta-clic> (**Sun**) ou <Alt-clic> (**HP-UX**) sur l'objet dont la couleur doit être copiée. L'outil Pinceau prend la forme d'une pipette et prend la couleur de l'objet sélectionné. Vous pouvez ensuite cliquer sur un autre objet avec l'outil Pinceau pour lui donner la couleur choisie.

Annulation

En plus de <Ctrl-Z> pour **Annuler** et <Ctrl-Maj-Z> pour **Revenir**, les logiciels en langage G peuvent définir le nombre d'actions à annuler ou à rétablir, de même qu'automatiquement mettre à jour les VIs appelant d'autres VIs lorsque le VI change d'interface. Pour plus d'informations, consultez la section *Annuler* du chapitre 7, *Personnalisation de votre environnement*.

Création de descriptions d'objets

Pour entrer une description pour un objet graphique, telle qu'une commande ou un indicateur, choisissez **Opérations sur les données» Description...** dans le menu local de l'objet. Vous devez être en mode d'édition pour modifier une description. Entrez la description dans la boîte de dialogue, présentée dans l'illustration suivante et cliquez sur **OK** pour l'enregistrer.



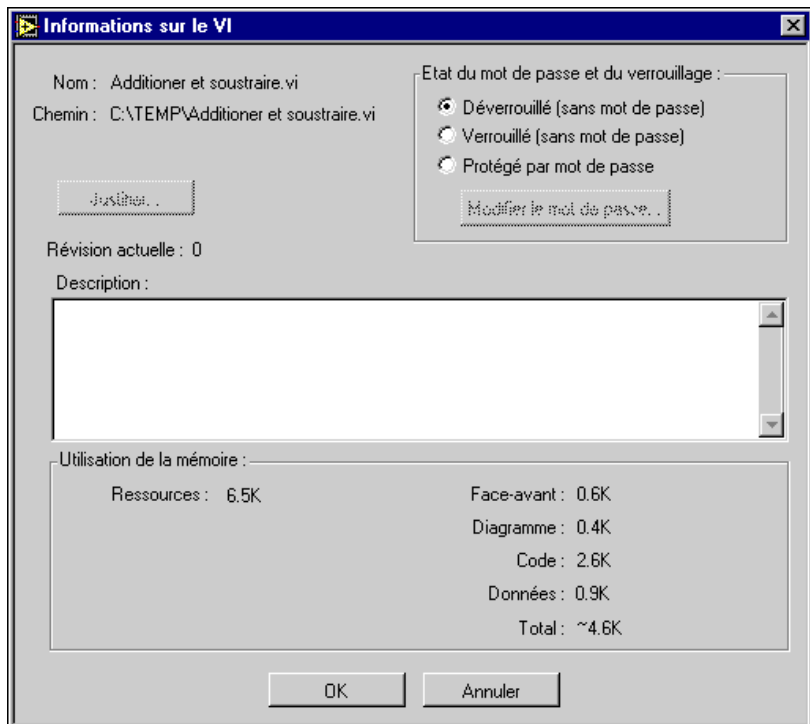
La description de l'objet apparaît également dans la fenêtre d'aide lorsque vous placez votre curseur sur l'objet. La meilleure façon d'accompagner les VIs que vous créez de leur propre "module d'aide" est d'entrer une description pour toutes les commandes et tous les indicateurs.

**Remarque**

Vous ne pouvez pas modifier les descriptions de sous-VIs depuis le diagramme d'un VI appelant. Vous pouvez modifier une description de VI via la commande Fenêtres»Infos sur le VI... lorsque la face-avant du VI est ouverte.

Création de descriptions pour les VIs

La sélection de **Fenêtres»Infos sur le VI...** affiche une fenêtre d'information pour le VI courant. Vous pouvez utiliser cette fenêtre pour réaliser les fonctions présentées dans l'illustration suivante.



- Entrer une description pour le VI. La fenêtre de description contient une barre de défilement pour les longues descriptions.
- Verrouiller ou déverrouiller le VI. Un VI verrouillé peut être exécuté mais pas modifié.
- Protéger le VI par mot de passe. Pour plus d'informations, consultez le chapitre 27, *Gestion de vos applications*.
- Visualiser le numéro de révision.

- Visualiser le chemin d'un VI.
- Visualiser la quantité de mémoire utilisée par un VI. La partie **Utilisation de la mémoire** de la fenêtre d'information affiche la quantité d'espace disque et de mémoire système utilisée par le VI (ce chiffre ne s'applique qu'à la quantité de mémoire utilisée par le VI et ne reflète pas la quantité de mémoire utilisée par ses sous-VIs).

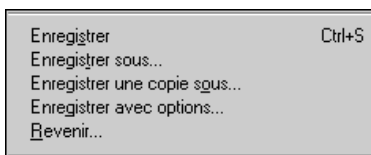
La mémoire utilisée est divisée entre l'espace nécessaire à la face-avant, au diagramme, au code du VI et aux données. L'utilisation de mémoire peut varier grandement, en particulier lors de la modification et de l'exécution du VI. Le diagramme utilise généralement le plus de mémoire. Lorsque vous n'apportez pas de modifications au diagramme, enregistrez le VI et fermez le diagramme pour libérer de l'espace mémoire pour les autres VIs. L'enregistrement et la fermeture de la face-avant libèrent également de la mémoire.

Enregistrement des VIs

Vous pouvez enregistrer les VIs en tant que fichiers individuels ou les regrouper et les enregistrer dans une bibliothèque de VIs.

Fichiers individuels

Cinq commandes du menu **Fichier** se rapportent à l'enregistrement des VIs en tant que fichier individuels.



Choisissez la commande **Enregistrer** pour enregistrer les modifications apportées à un VI existant. S'il s'agit d'un nouveau VI, une boîte de dialogue apparaît, invitant à entrer un nom et un emplacement dans lequel enregistrer le VI.

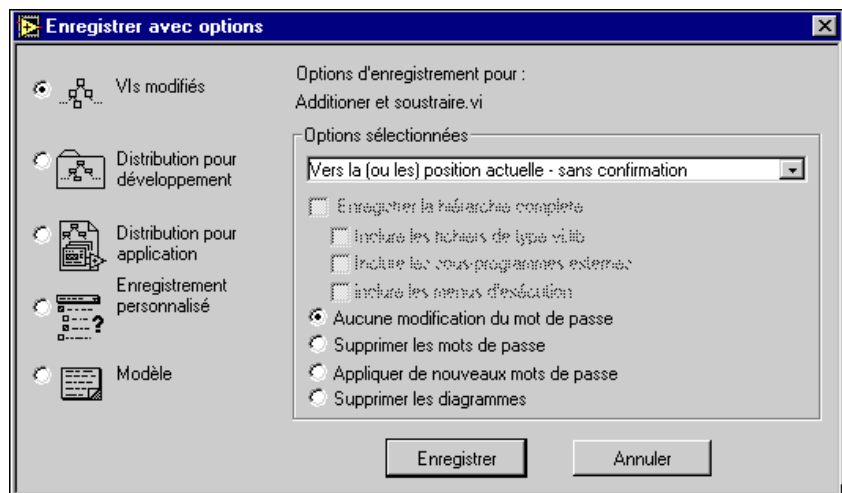
Pour enregistrer un VI sous un nouveau nom, vous pouvez utiliser **Enregistrer sous...**, **Enregistrer une copie sous...**, ou **Enregistrer avec options...** dans le menu **Fichier**.

Lorsque vous choisissez la commande **Enregistrer sous...**, une copie du VI en mémoire est enregistrée sur le disque avec le nouveau nom choisi. Une fois l'enregistrement réalisé, le VI en mémoire pointe vers la nouvelle

version. Tous les VIs appelants de la version précédemment en mémoire du VI appellent maintenant le nouveau VI. Si vous entrez un nouveau nom pour le VI, la version sur disque du VI d'origine n'est ni écrasée ni supprimée.

Lorsque vous choisissez la commande **Enregistrer une copie sous...**, une copie du VI en mémoire est enregistrée sur le disque avec le nouveau nom choisi. Cela n'affecte pas le nom du VI en mémoire.

Enregistrer avec options... entraîne l'apparition d'une boîte de dialogue dans laquelle vous pouvez choisir d'enregistrer une hiérarchie complète sur le disque, avec ou sans mot de passe ou sans diagramme. Cette option est particulièrement utile pour distribuer des VIs ou faire des copies de sauvegarde.



Pour plus d'informations sur cet élément de menu, consultez le chapitre 27, *Gestion de vos applications*.



Avertissement

Vous ne pouvez pas modifier un VI après l'avoir enregistré sans son diagramme. Faites toujours une copie de l'original.

Vous pouvez utiliser la commande **Fichier»Revenir...** pour revenir à la dernière version enregistrée du VI sur lequel vous travaillez. Une boîte de dialogue apparaît, invitant à confirmer la suppression des changements apportés au VI.

Les VIs modifiés depuis leur dernier enregistrement sont identifiés par un astérisque dans leur barre de titre et dans la liste des VIs ouverts affichée dans le menu **Fenêtres**. Lorsque vous enregistrez un VI, l'astérisque disparaît, jusqu'au prochain changement.

Pour plus d'informations sur l'enregistrement de copies de sauvegarde, consultez le chapitre 27, *Gestion de vos applications*.



Avertissement

*N'enregistrez pas vos VIs dans le répertoire `vi.lib` du répertoire de LabVIEW ou de BridgeVIEW. National Instruments met ce répertoire à jour lors des nouvelles versions de LabVIEW ou de BridgeVIEW. Le fait de placer des VIs dans `vi.lib` entraîne des risques de conflit au cours des futures installations. Si vous voulez que vos VIs apparaissent dans la palette **Fonctions**, consultez la section *Personnalisation des palettes Commandes et Fonctions* du chapitre 7, *Personnalisation de votre environnement*.*

Bibliothèques de VIs (.LLB)

Vous pouvez regrouper plusieurs VIs et les enregistrer dans une bibliothèque. A moins d'avoir une bonne raison de procéder ainsi, il est toujours préférable de les enregistrer en tant que fichiers individuels, organisés en répertoires. La liste suivante vous aidera à décider de la méthode appropriée.

Raisons pour enregistrer des VIs sous forme de bibliothèques :

- Si vous utilisez Microsoft Windows 3.1 ou pensez à transférer vos fichiers sous Windows 3.1, l'enregistrement de VIs dans des bibliothèques (fichiers `.llb`) vous permet de donner des noms composés de 255 caractères maximum à vos fichiers. Les autres systèmes d'exploitation supportent les longs noms de fichiers (31 caractères pour Macintosh, 255 caractères sous Windows 95/NT et les systèmes UNIX).
- Si vous pensez à transférer vos VIs sur d'autres plates-formes, le transfert peut s'avérer plus simple sous la forme de bibliothèques plutôt que de fichiers individuels.
- Si vous avez besoin d'espace disque, les fichiers enregistrés dans des bibliothèques étant compressés, ils utilisent moins d'espace mémoire.

Raisons pour enregistrer des VIs sous forme de fichiers individuels :

- Si vous conservez vos VIs en tant que fichiers individuels, vous pouvez les gérer à l'aide du système de fichiers (copie, déplacement, modification du nom, sauvegarde, gestion du code source).
- Vous ne pouvez pas avoir de hiérarchie dans les bibliothèques de VIs, celles-ci ne contenant pas de sous-répertoires ou dossiers.
- Le chargement et l'enregistrement des fichiers sont plus rapides depuis le système de fichiers que des bibliothèques de VIs. Moins d'espace disque est nécessaire pour les fichiers temporaires pendant le chargement et l'enregistrement.
- Le stockage des VIs et commandes dans des fichiers individuels est plus pratique que le stockage d'un projet entier dans le même fichier.
- Les bibliothèques de VIs ne sont pas compatibles avec les commandes de code source du *Kit du développeur professionnel*.



Remarque

Beaucoup de VIs accompagnant LabVIEW ou BridgeVIEW sont stockés dans des bibliothèques. Cette méthode rend les emplacements de stockage plus cohérents et indépendants de la plate-forme.

Création de bibliothèques de VIs

Pour créer une bibliothèque de VIs :

1. Choisissez **Enregistrer sous...** ou **Enregistrer une copie sous...** dans le menu **Fichier**.
2. (**Macintosh**) Si vous configurez, grâce à l'option **Préférences»Divers**, l'utilisation des boîtes de dialogue d'origine, cliquez sur **Utiliser les LLB** dans la boîte de dialogue qui apparaît.
3. Dans la boîte de dialogue qui apparaît à la suite de l'étape 1 ou de l'étape 2, cliquez sur **Nouveau...** ou **Nouvelle bibliothèque de VIs**.
4. Dans la boîte de dialogue qui apparaît, entrez le nom de la nouvelle bibliothèque et cliquez sur **Bibliothèque de VIs**. Le logiciel ajoute automatiquement une extension **.llb** au nom de la nouvelle bibliothèque (si vous ne le faites pas de vous-même).
5. Une boîte de dialogue apparaît, vous invitant à nommer le VI et à l'enregistrer dans la nouvelle bibliothèque.

Enregistrement dans des bibliothèques de VIs existantes

Vous pouvez enregistrer de nouveaux VIs dans une bibliothèque comme s'il s'agissait d'un répertoire. Une fois une des commandes d'enregistrement du menu **Fichier** choisie, le nom du fichier bibliothèque, avec l'extension `.llb`, apparaît dans la boîte de dialogue (les utilisateurs du Macintosh utilisant la boîte de dialogue d'origine doivent d'abord cliquer sur **Utiliser les LLB** avant de sélectionner une bibliothèque). Lorsque vous double-cliquez sur le nom du fichier bibliothèque ou cliquez sur **Ouvrir**, la boîte de dialogue présente un élément vous permettant d'enregistrer le fichier dans la bibliothèque.

Modification du contenu des bibliothèques

Vous pouvez supprimer un fichier d'une bibliothèque à l'aide de la commande **Fichier»Editer une bibliothèque de VIs...** Lorsque la boîte de dialogue **Editer une bibliothèque de VIs** apparaît, vous pouvez également définir les fichiers comme étant de **Niveau principal**. Cette fonction a deux utilités : premièrement, lorsque vous créez une application à l'aide des bibliothèques de développement d'application, les VIs de **Niveau principal** s'ouvrent automatiquement lors de l'exécution de l'application ; deuxièmement, lorsque vous double-cliquez sur une bibliothèque spécifique du système de fichiers sous Windows ou sur Macintosh, ou que vous lancez votre environnement de développement graphique avec un nom de bibliothèque spécifié dans la ligne de commande sous Windows ou UNIX, le logiciel ouvre automatiquement tous les VIs de niveau supérieur.

Les VIs de niveau supérieur sont également séparés par une ligne de tous les autres VIs de la bibliothèque lorsque vous visualisez la boîte de dialogue représentant le contenu d'un fichier `.llb`.

Utilisation des sous-VIs

Ce chapitre présente le concept de l'organisation hiérarchique des applications en G et suggère deux méthodes de création des sous-VIs. Vous y trouverez également une description de deux utilitaires : la fenêtre Hiérarchie, qui affiche la hiérarchie de vos VIs et l'utilitaire Rechercher, qui recherche les occurrences de sous-VIs, de même que tous les autres objets ou chaînes de texte indiqués.

Conception hiérarchique

Deux des points-clés de la création d'applications en G sont la compréhension et l'utilisation de la nature hiérarchique d'un VI. Une fois un VI créé, il peut être utilisé comme sous-VI dans le diagramme d'un VI de niveau supérieur. Un sous-VI est donc analogue à un sous-programme en C. Tout comme il n'existe aucune limite au nombre de sous-programmes pouvant être utilisés dans un programme écrit en C, il n'existe aucune limite au nombre de VIs pouvant être utilisés dans un programme écrit en G. Un sous-VI peut également être appelé au sein d'un autre sous-VI.

Lors de la création d'une application, vous devez commencer par le VI principal et définir les entrées et les sorties de l'application. Vous devez ensuite utiliser d'autres VIs en tant que sous-VIs permettant de réaliser les opérations nécessaires sur les données lors de leur passage dans le diagramme. Si un diagramme contient un grand nombre d'icônes, vous pouvez grouper ces icônes dans un VI de niveau inférieur afin de conserver la simplicité du diagramme. Cette approche modulaire rend les applications plus faciles à mettre au point, à comprendre et à maintenir.

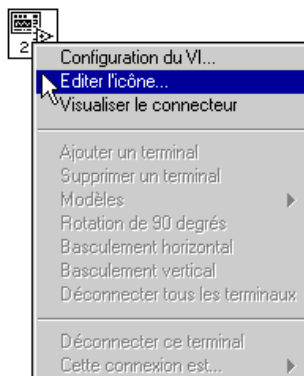
Vous pouvez créer un sous-VI à partir d'un VI ou créer un sous-VI à partir d'une sélection (portion) d'un VI. Ces deux méthodes sont présentées dans ce chapitre.

Création de sous-VIs à partir de VIs

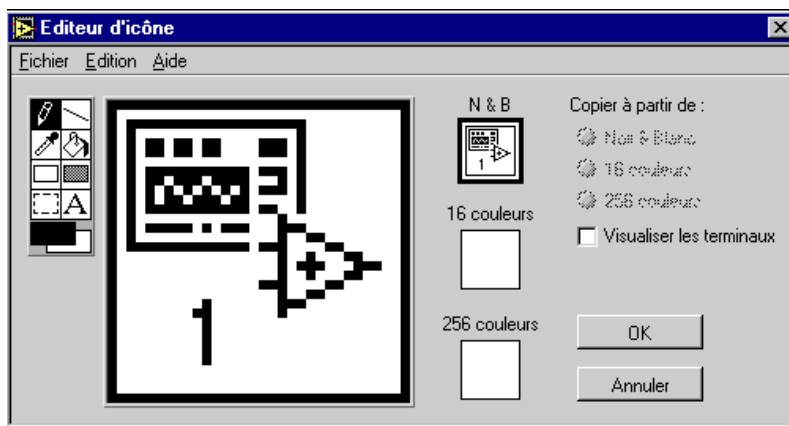
L'icône d'un VI en est sa représentation graphique. Le *connecteur* d'un VI affecte les commandes et indicateurs aux terminaux d'entrée et de sortie. Pour appeler votre VI à partir du diagramme d'un autre VI, il faut d'abord créer une icône et un connecteur. Cette section présente la création et la modification d'une icône et du connecteur d'un VI.

Création de l'icône

Pour créer une icône, double-cliquez sur l'icône dans l'angle supérieur droit de la face-avant ou cliquez avec le bouton droit de la souris sur l'icône et choisissez **Editer l'icône...** dans le menu déroulant, tel que présenté dans l'illustration suivante.



Une fois la commande **Editer l'icône...** sélectionnée, l'Editeur d'icônes apparaît.



Utilisez les outils à gauche de la fenêtre, présentée dans l'illustration précédente, pour créer l'icône dans la zone de pixels agrandis. L'image en taille réelle apparaît dans la zone correspondante, sur la droite de la zone d'édition.

Suivant le type de votre moniteur, vous pouvez dessiner une icône différente pour l'affichage en mode monochrome, 16 couleurs et 256 couleurs, chaque version étant dessinée et enregistrée séparément. La valeur par défaut de l'éditeur est **Noir & Blanc**, mais peut être changée en cliquant sur les autres commandes de couleurs. Vous pouvez copier d'une icône en couleur vers une icône en noir et blanc, et vice versa, en choisissant la commande **Copier à partir de**, située sur la droite de l'Editeur d'icône.



Remarque

Il est toujours judicieux de créer au moins une icône en noir & blanc. Si vous ne créez qu'une icône en couleur, elle ne sera pas affichée si vous l'ajoutez à la palette Fonctions. Elle ne pourra pas être imprimée et ne s'affichera pas sur un écran noir & blanc. Si un VI n'a pas d'icône noir & blanc, le logiciel utilise une icône vierge. Pour plus d'informations, consultez la section Personnalisation des palettes Commandes et Fonctions du chapitre 7, Personnalisation de votre environnement.

Les outils situés sur la gauche de la zone d'édition remplissent les fonctions suivantes :



Outil Crayon : dessine et efface pixel par pixel. Utilisez la touche <Maj> pour restreindre le dessin à des lignes horizontales ou verticales.



Outil Ligne : dessine des lignes droites. Utilisez la touche <Maj> pour restreindre le dessin à des lignes horizontales, verticales ou diagonales.



Outil Pipette : sélectionne une couleur d'un élément de l'icône pour le premier plan. Utilisez la touche <Maj> pour sélectionner la couleur d'arrière-plan avec la pipette.



Outil Pot de peinture : remplit une zone délimitée avec la couleur de premier plan.



Outil Rectangle : dessine une bordure rectangulaire avec la couleur de premier plan. Double-cliquez pour entourer l'icône avec la couleur de premier plan. Utilisez la touche <Maj> pour contraindre le rectangle en un carré.



Outil Rectangle plein : dessine un rectangle entouré de la couleur de premier plan et rempli avec la couleur d'arrière-plan. Double-cliquez pour entourer l'icône avec la couleur de premier plan et la remplir avec la couleur d'arrière-plan. Utilisez la touche <Maj> pour contraindre le rectangle en un carré.



Outil Marquise : sélectionne une zone de l'icône à déplacer, copier ou supprimer. Double-cliquez pour sélectionner l'icône toute entière. Le fait d'appuyer sur la touche <Supprimer> efface la portion sélectionnée de l'icône. Utilisez la touche <Maj> pour contraindre le rectangle en un carré.



Outil Texte : entre du texte dans l'icône. Double-cliquez sur cette icône pour sélectionner une police différente. Utilisez les touches de curseur pour déplacer le texte dans l'icône.



Outil Premier plan/arrière-plan : affiche les couleurs de premier plan et d'arrière-plan courantes. Cliquez sur les deux parties pour obtenir une palette permettant de choisir une nouvelle couleur.

Le fait de maintenir la touche <Ctrl> (**Windows**), <option> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) temporairement enfoncée change tous les outils en une outil Pipette.

Les boutons en bas à droite de l'éditeur sont les suivants :

OK : enregistre le dessin comme icône de VI et retourne à la face-avant.

Annuler : fait revenir à la face-avant sans enregistrer les modifications.



Remarque

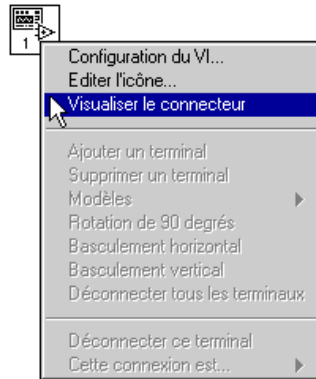
*Vous pouvez utiliser le menu **Edition** pour couper, copier et coller des images depuis et vers l'icône. Lorsque vous collez une image et qu'une portion de l'icône est sélectionnée, l'image est redimensionnée pour tenir dans la sélection.*

Définition des terminaux des connecteurs

Vous pouvez envoyer et recevoir les données depuis un sous-VI grâce aux terminaux de son cadre connecteur. Vous devez définir les connexions en choisissant le nombre de terminaux à donner au VI et en affectant une commande ou un indicateur de face-avant à chacun de ces terminaux. Seuls les commandes et indicateurs utilisés par la programmation et connectés au sous-VI nécessitent des terminaux dans le cadre connecteur.

Si le connecteur de votre VI n'est pas déjà affiché dans l'angle supérieur droit de la face-avant, choisissez l'option **Visualiser le connecteur** dans le

menu local de l'icône, tel que présenté dans l'illustration suivante. Le cadre connecteur ne peut pas être visualisé dans la fenêtre du diagramme.

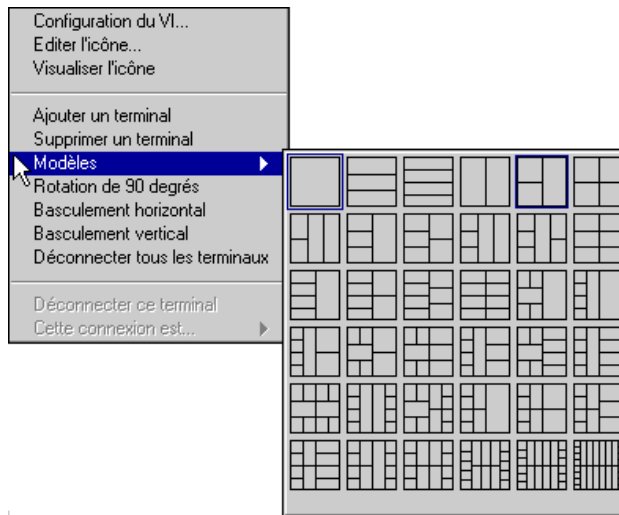


Le connecteur remplace l'icône dans l'angle supérieur droit de la face-avant. La sélection initiale est un terminal avec autant de terminaux à gauche du cadre connecteur que de commandes sur la face-avant et autant de terminaux à droite du cadre connecteur que d'indicateurs sur la face-avant. Si cela n'est pas possible, le logiciel choisit la correspondance la plus proche.

Chaque rectangle sur le connecteur représente une zone du terminal que vous pouvez utiliser pour une entrée ou une sortie pour le VI. Si vous souhaitez utiliser un terminal différent pour le VI, vous pouvez en sélectionner un différent.

Sélection et modification de terminaux

Pour sélectionner un terminal différent pour votre VI, cliquez avec la souris sur le connecteur et choisissez **Modèles** dans le menu local.



Une bordure solide met en évidence le terminal associé à l'icône. N'importe quel terminal peut être sélectionné si vous souhaitez en changer.

Pour ajouter un terminal à la liste, placez le curseur à l'endroit auquel l'ajouter, cliquez avec la souris et choisissez **Ajouter un terminal dans le menu local**.

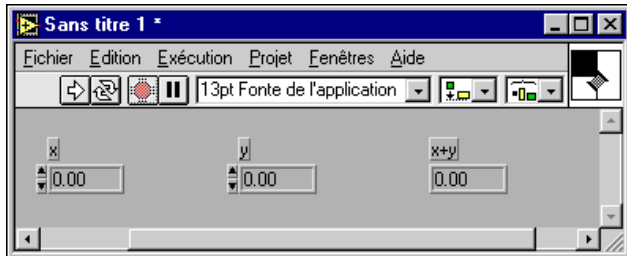
Pour supprimer un terminal de la liste, placez le curseur sur le terminal à supprimer, cliquez avec la souris et choisissez **Supprimer un terminal dans le menu local**.

Pour modifier la position des terminaux au niveau du connecteur, choisissez une des commandes suivantes dans le menu local du cadre connecteur : **Basculement horizontal, Basculement vertical ou Rotation de 90 degrés**.

Affectation de terminaux aux commandes et indicateurs

Une fois le type de terminal choisi pour votre connecteur, vous devez affecter des commandes et indicateurs de face-avant à vos terminaux. La procédure est la suivante :

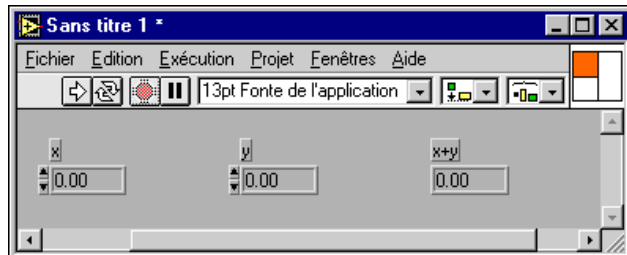
1. Cliquez dans un terminal du connecteur. L'outil devient automatiquement l'outil Bobine. Le terminal devient noir.



2. Cliquez sur la commande ou sur l'indicateur de face-avant à affecter au terminal sélectionné. Une marquée encadre la commande sélectionnée.



3. Placez le curseur dans une zone vide de la face-avant et cliquez avec le bouton gauche de la souris. La marquée disparaît et le terminal prend la couleur des données de la commande connectée, indiquant que le terminal a bien été affecté.





Remarque

Si le terminal du connecteur devient blanc, cela signifie qu'une connexion n'a pas été effectuée. Reprenez les étapes 1 à 3 jusqu'à ce que le terminal de connecteur prenne la couleur correspondant aux données manipulées.

Bien que vous utilisiez l'outil Bobine pour affecter des terminaux aux connecteurs des commandes et indicateurs de face-avant, aucun fil de liaison n'apparaît entre le connecteur et ses commandes et indicateurs.

4. Reprenez les étapes 1 et 2 pour chaque commande et indicateur à connecter.

Vous pouvez également sélectionner la commande ou l'indicateur avant de sélectionner le terminal. Vous pouvez choisir un motif avec plus de terminaux que nécessaire et laisser quelques terminaux sans connexion si vous pensez effectuer des changements sur le VI dans le futur et ajouter des entrées ou des sorties. Le fait d'avoir quelques connexions supplémentaires disponibles signifie que les nouvelles entrées ou sorties ne perturbent pas les VIs utilisant déjà ce VI comme sous-VI. Les terminaux non affectés ne perturbent pas le fonctionnement du VI. Vous pouvez avoir plus de commandes de face-avant que de terminaux.

Le cadre connecteur contient, au maximum, 28 terminaux. Si votre face-avant contient plus de 28 commandes et indicateurs que vous voulez utiliser comme entrées ou sorties, groupez en quelques-uns dans un cluster et affectez ce cluster à un terminal du connecteur. Pour plus d'informations, consultez la section *Clusters* du chapitre 14, *Commandes et indicateurs de tableau et de cluster*.

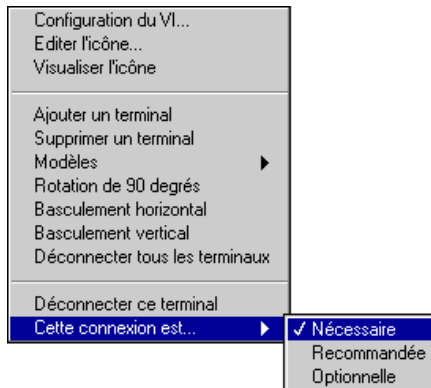
Connexions nécessaires, recommandées et optionnelles pour les sous-VIs

Le G possède une fonction vous permettant de ne pas oublier de lier les connexions de sous-VI : des indications des connexions nécessaires, recommandées ou optionnelles dans le cadre connecteur et les mêmes indications dans la fenêtre d'aide.

Les entrées et sorties des VIs de la bibliothèque `vi.lib` peuvent être qualifiées de **Nécessaires**, **Recommandées** ou **Optionnelles**. Le G donne aux entrées ou sorties que vous créez la valeur **Recommandée** par défaut.

Lorsqu'une entrée est identifiée par **Nécessaire**, le VI ne peut pas être exécuté comme sous-VI si cette entrée n'est pas câblée correctement. Lorsqu'une entrée ou sortie est identifiée comme étant recommandée, le VI peut être exécuté mais la fenêtre **Liste des erreurs** génère une mise en garde si cette fonction est activée.

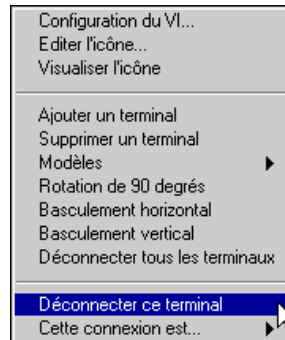
Pour voir si les connexions sont **Nécessaires**, **Recommandées** ou **Optionnelles** ou pour leur donner une de ces identifications, cliquez sur un terminal dans le cadre connecteur et choisissez un élément du sous-menu **Cette connexion est**. Une marque indique son état, tel que présenté dans l'illustration suivante.



Dans la fenêtre d'aide, les connexions Nécessaires apparaissent en gras, les connexions Recommandées en texte normal et les connexions Optionnelles en texte gris.

Suppression des connexions de terminaux

Vous pouvez supprimer des connexions entre terminaux et les commandes ou indicateurs qui leur correspondent individuellement ou en une seule fois. Pour supprimer une connexion particulière, cliquez avec la souris sur le terminal à déconnecter au niveau du connecteur et choisissez l'option **Déconnecter ce terminal** dans le menu local, tel que présenté dans l'illustration suivante.



Le terminal devient blanc, indiquant que la connexion sélectionnée n'existe plus. Pour supprimer toutes les connexions du connecteur, cliquez avec la souris n'importe où sur le connecteur et choisissez l'option **Déconnecter tous les terminaux** dans le menu local. Remarquez que cette action est différente de **Supprimer le terminal** étant donné que **Déconnecter ce terminal** ne supprime pas le terminal de la liste de terminaux.

Confirmation des connexions de terminaux

Pour voir les commandes ou indicateurs affectés à un terminal particulier, cliquez sur une commande, un indicateur ou un terminal avec l'outil Bobine lorsque le cadre connecteur est visible. Le logiciel sélectionne l'objet correspondant.

Création de sous-VIs à partir d'une partie d'un VI

Vous pouvez convertir une portion d'un VI en un sous-VI pouvant être appelé par un autre VI. Sélectionnez une section d'un VI, choisissez l'option **Editer»Créer un sous-VI** et la section devient un sous-VI. Des commandes et indicateurs sont automatiquement créés pour le nouveau sous-VI, le sous-VI est automatiquement lié aux fils de liaison existants et une icône de sous-VI remplace la section sélectionnée du diagramme dans le VI d'origine.

La création d'un sous-VI à partir d'une partie de VI revient au même que de supprimer les objets sélectionnés et de les remplacer par un sous-VI, à l'exception des comportements suivants :

- Aucun des terminaux de face-avant inclus dans la sélection n'est supprimé du VI appelant. Les terminaux de face-avant sont conservés sur le VI appelant et sont connectés au sous-VI.
- Tous les attributs inclus dans la sélection sont conservés sur le VI appelant et sont connectés au sous-VI. Les attributs sélectionnés sont remplacés par les terminaux de face-avant dans le sous-VI, qui agissent comme des canaux transférant la valeur d'un attribut vers l'intérieur et l'extérieur du sous-VI.
- Lorsqu'une sélection comprend une variable locale, elle est remplacée par un terminal de face-avant dans le sous-VI. La variable locale est conservée dans le VI appelant et est connectée au sous-VI. Lorsque plusieurs instances d'une même variable locale sont sélectionnées, la première instance devient un terminal de face-avant et les autres restent des variables locales dans le sous-VI. Le reste des instances sélectionnées est supprimé du VI appelant et inclus dans le sous-VI.
- Etant donné que les variables locales peuvent à la fois lire et écrire dans un terminal de face-avant, lorsque des instances de variables locales en lecture et écriture sont sélectionnées, deux objets de face-avant sont créés dans le sous-VI : un pour passer la valeur de la variable locale au sous-VI et un autre pour faire sortir la valeur du sous-VI. Le terminal de face-avant créé pour représenter les variables locales en lecture a le suffixe `lecture` ajouté à son nom alors que le terminal de face-avant créé pour représenter les variables locales en écriture a le suffixe `écriture` ajouté à son nom.

Règles et recommandations

La possibilité de créer un sous-VI à partir d'une sélection peut être très pratique, sans pour autant être aussi simple qu'il n'y paraît. Une planification minutieuse est nécessaire à la création d'une hiérarchie logique de VIs. Vous devez également décider des objets à inclure dans la sélection afin d'éviter des situations où les fonctionnalités des VIs résultants sont modifiées. En cas de problème, aucun sous-VI n'est créé à partir de la sélection, mais une boîte de dialogue apparaît, expliquant la raison de l'échec de l'opération. Dans les cas où un problème potentiel existe, une boîte de dialogue apparaît avec une explication du problème potentiel. C'est ensuite à vous de décider si vous voulez continuer.



Remarque

S'il n'existe aucune variable locale ou attribut node dans une sélection, vous pouvez utiliser une structure Séquence pour encapsuler le code à envoyer au sous-VI et avoir un aperçu du résultat.

Les règles et recommandations suivantes peuvent aider à l'utilisation de ces fonctions.

Nombre de connexions

N'effectuez pas de grandes sélections qui pourraient créer un sous-VI avec plus de 28 entrées ou sorties. Ce nombre est le nombre maximum de terminaux sur un cadre connecteur.

Rappelez-vous que chaque terminal de face-avant, chaque attribut et certaines variables locales contenus dans la sélection ont besoin d'un emplacement dans le cadre connecteur ; une sélection avec un grand nombre de ces éléments peut entraîner un manque d'emplacements.

Pour éviter de dépasser le nombre maximum de terminaux, sélectionnez une section plus réduite du diagramme ou groupez les données dans des tableaux ou clusters avant de sélectionner une région du diagramme à convertir.

Cycles

Évitez de créer des sélections créant des cycles dans le diagramme. Les cycles ont lieu lorsqu'un flux de données provient d'une sortie du sous-VI et se termine à une entrée du sous-VI.

Identifier des cycles pendant la sélection est une chose difficile, mais le G le détecte pour vous. Lorsque le G détecte un cycle, une boîte de dialogue est affichée, vous invitant à créer un nouveau VI à partir de la sélection. Si vous choisissez de créer un nouveau VI, un nouveau VI sans titre est créé à

partir de la sélection. Les éléments sélectionnés dans le diagramme d'origine restent inchangés.

Attribute Nodes dans les boucles

N'incluez pas d'attribute nodes à l'intérieur d'une boucle dans votre sélection.

Un attribute node étant retenu dans le VI appelant et connecté au sous-VI, l'exécution du sous-VI ne met pas à jour la valeur de l'attribut à chaque itération de la boucle. Il est impossible de créer un sous-VI dans de tels cas.

Sélections illogiques

Ne convertissez pas des sélections en un sous-VI lorsque cela n'a pas de sens. Par exemple, il est illogique de convertir une sélection consistant en un objet dans une structure Séquence et un autre objet au-dehors de la structure Séquence sans inclure la structure Séquence elle-même.

De telles sélections entraînent un affichage de l'explication du problème.

Variables locales et terminaux de face-avant dans les boucles

Essayez d'éviter d'inclure des variables locales ou des terminaux de face-avant à l'intérieur d'une boucle dans votre sélection.

Les terminaux de face-avant ou variables locales sélectionnés étant conservés sur le VI appelant et connectés au sous-VI, l'exécution du sous-VI ne met pas à jour la valeur de ces éléments à chaque itération de la boucle. Cela peut entraîner une modification de la fonctionnalité du VI appelant.

Dans ce cas, un message apparaît vous demandant si vous voulez continuer ou annuler l'opération.

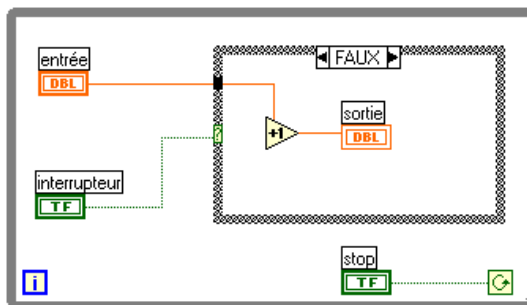
Structures Condition contenant des Attribute Nodes, variables locales ou terminaux de face-avant

Essayez d'éviter d'inclure, dans votre sélection, une structure Condition contenant un attribute node, un terminal de face-avant ou une variable locale configurée pour recevoir une valeur.

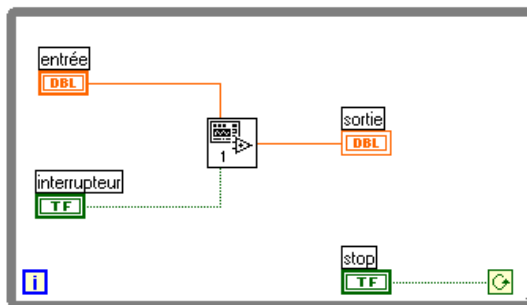
Dans un tel cas, lorsque le sous-VI est exécuté, une valeur spécifique doit toujours être écrite vers l'objet, car celui-ci est connecté au sous-VI. Bien qu'elle existe dans le diagramme d'origine, une valeur ne peut être écrite que si la partie VRAI ou FAUX de la structure Condition est exécutée. Cela peut entraîner une modification de la fonctionnalité du VI appelant.

Un message apparaît alors, vous demandant si vous voulez continuer ou annuler l'opération. Si vous continuez, vous devez d'abord modifier le sous-VI pour qu'il fournisse une valeur dans toutes les conditions.

L'illustration suivante est un exemple de la situation décrite ci-dessus.



La structure Condition est sélectionnée pour être convertie en sous-VI. Le terminal de sortie de la face-avant n'est pas inclus dans le diagramme d'origine et est connecté au sous-VI via un connecteur, tel que présenté dans l'illustration suivante.



Dans le diagramme d'origine, le G écrit une valeur vers le terminal de sortie uniquement lorsque la partie FAUX de la condition est exécutée. Lorsque la sélection devient un sous-VI, le G écrit toujours une valeur vers le terminal de sortie que ce soit la partie VRAI ou FAUX de la condition qui soit exécutée dans le sous-VI. Vous devez modifier le sous-VI pour qu'il fournisse des données pour les conditions supplémentaires.

Utilisation de la fenêtre Hiérarchie

La fenêtre Hiérarchie contient une représentation graphique de la hiérarchie appelante pour tous les VIs en mémoire, y compris les définitions de type et les variables globales.

Vous pouvez configurer différents aspects de l'affichage dans la fenêtre Hiérarchie. Par exemple, vous pouvez afficher la représentation horizontalement ou verticalement et inclure ou exclure les VIs du répertoire `vi.lib`, les variables globales ou les définitions de type.

D'autres fonctions pratiques de la fenêtre Hiérarchie sont par exemple la possibilité d'accéder aux options **Configuration du VI...**, **Editer l'icône...**, **Infos sur le VI...** et **Imprimer la documentation...**, la possibilité de faire glisser ou de copier et coller des nœuds de hiérarchie dans le diagramme d'un autre VI comme sous-VI et la possibilité de rechercher des nœuds de hiérarchie par leur nom.

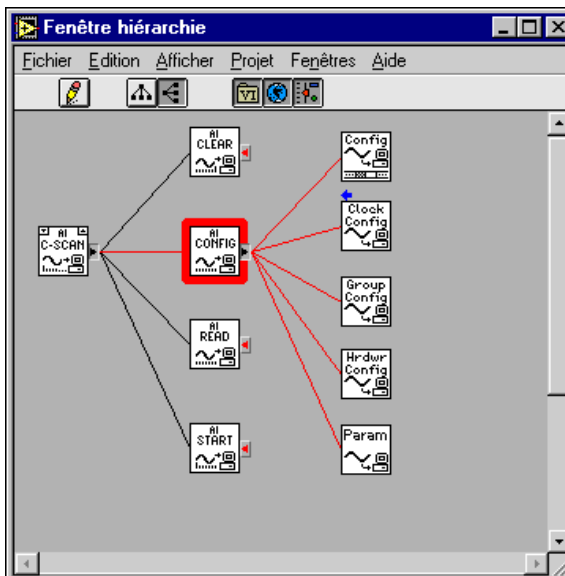
Ouverture de la fenêtre Hiérarchie

Vous pouvez ouvrir la fenêtre Hiérarchie d'une des façons suivantes :

- Choisissez l'option **Projet** » **Visualiser la hiérarchie des VIs**. La fenêtre Hiérarchie est affichée avec le VI de la fenêtre active comme nœud principal.
- Choisissez l'option **Visualiser la hiérarchie des VIs** dans les menus locaux des sous-VIs, variables globales ou définitions de type. La fenêtre Hiérarchie est affichée avec le sous-VI, la variable globale ou la définition de type comme nœud principal.
- Si la fenêtre Hiérarchie est déjà ouverte, vous pouvez la faire passer au premier plan en la sélectionnant dans la liste des fenêtres ouvertes dans le menu **Fenêtres**.

Vous pouvez passer de l'affichage vertical à horizontal grâce à une commande du menu **Afficher** ou en cliquant sur le bouton **Présentation horizontale** ou **Présentation verticale** en haut de la fenêtre.

Dans un affichage horizontal, les sous-VIs sont présentés à droite de leur VI appelant alors que dans un affichage vertical, ils sont présentés en dessous. Les sous-VIs sont toujours connectés par des lignes à leurs VIs appelants. La fenêtre présentée dans l'illustration suivante est affichée horizontalement.



Les boutons fléchés et les flèches près des nœuds indiquent ce qui est affiché et ce qui est masqué, comme suit :



- Un bouton fléché rouge pointant vers le nœud indique que quelques VIs ou certains d'entre eux sont masqués. Cliquer sur le bouton affiche les sous-VIs immédiats du nœud.



- Un bouton fléché noir pointant vers les sous-VIs du nœud indique que tous les sous-VIs immédiats sont affichés.



- Une flèche bleue pointant vers les VIs appelants du nœud indique que le nœud a quelques appelants supplémentaires qui ne sont pas affichés.

Si un nœud n'a aucun sous-VI, aucun bouton fléché rouge ou noir n'est affiché.

Un nœud devient le nœud principal, entouré d'une bordure rouge épaisse, lorsqu'une opération est réalisée dans ce nœud. Il perd cette mise en évidence dès qu'une opération est effectuée dans un autre nœud.

Lorsque vous placez votre curseur sur des objets de la fenêtre Hiérarchie et que votre curseur s’immobilise sur un nœud, le G affiche le nom du nœud en dessous de son icône. Si vous préférez, vous pouvez utiliser une commande du menu **Afficher** pour afficher le chemin complet au lieu du nom.

Vous pouvez double-cliquer sur n’importe quelle icône de la fenêtre Hiérarchie pour ouvrir le VI associé.

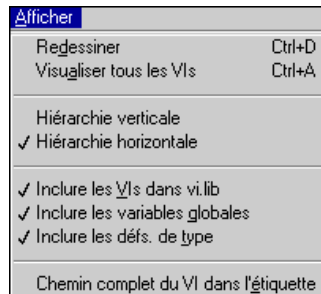
Si vous affichez la fenêtre d’aide et placez le curseur sur une icône, le G affiche l’information de fenêtre d’aide pour ce VI.

Options de la fenêtre Hiérarchie

Plusieurs options vous permettent de contrôler l’affichage de la fenêtre Hiérarchie ; d’autres vous permettent de réaliser des actions sur les nœuds affichés dans la fenêtre. Ces options sont accessibles dans le menu **Afficher**, par des boutons situés en haut de la fenêtre Hiérarchie, dans le menu local des nœuds et grâce à la souris. Vous pouvez également accéder au menu **Afficher** en cliquant avec le bouton droit de la souris sur un espace vide de la fenêtre Hiérarchie.

Options du menu Afficher

Le menu **Afficher** contient les éléments suivants liés à l’affichage de la fenêtre Hiérarchie. Plusieurs de ces éléments sont également accessibles par des boutons situés en haut de la fenêtre Hiérarchie.



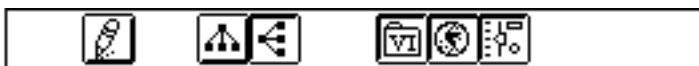
- **Redessiner** : réorganise l’affichage de la fenêtre afin de réduire au maximum les intersections de lignes et optimiser la symétrie, une option utile à la suite de quelques opérations sur les nœuds de hiérarchie. Si un nœud principal existe, la fenêtre se déplace pour

permettre de le visualiser. Dans le cas contraire, la fenêtre se déplacera pour montrer la première racine des sous-VIs.

- **Visualiser tous les VIs** : affiche tous les VIs masqués. Aucun nœud principal n'existe après la sélection de cet élément. Cela n'affecte pas les autres paramètres du menu **Afficher**. En d'autres termes, les VIs de `vi.lib`, les variables globales et les définitions de type non inclus dans la hiérarchie restent masqués.
- **Hiérarchie verticale** : arrange les nœuds de haut en bas avec les VIs appelants au-dessus de leurs sous-VIs.
- **Hiérarchie horizontale** : arrange les nœuds de gauche à droite avec les VIs appelants à gauche de leurs sous-VIs.
- **Inclure les VIs de `vi.lib`** : inclut ou exclut les VIs provenant du répertoire `vi.lib`.
- **Inclure les variables globales** : inclut ou exclut les variables globales.
- **Inclure les défs. de type** : inclut ou exclut les définitions de type.
- **Chemin complet du VI dans l'étiquette** : affiche le chemin complet du VI ou simplement son nom dans l'info-bulle pour chaque nœud de hiérarchie.

Boutons de barre d'outils de hiérarchie

Une barre d'outils dans la fenêtre Hiérarchie contient des boutons affectant l'affichage de la fenêtre.



Ces boutons, qui réalisent certaines des actions des éléments du menu **Afficher**, sont les suivants :



Refaire le schéma : réorganise l'affichage de la fenêtre pour réduire les intersections entre les lignes et maximiser la symétrie, ce qui peut s'avérer utile après plusieurs opérations sur des nœuds de hiérarchies. Si un nœud principal existe, la fenêtre se déplace pour afficher ce nœud. Dans le cas contraire, la fenêtre se déplace pour afficher la première racine de sous-VIs.



Présentation verticale : arrange les nœuds de haut en bas, avec les VIs appelants au-dessus de leurs sous-VIs.



Présentation horizontale — arrange les nœuds de gauche à droite, avec les VIs appelants à gauche de leurs sous-VIs.



Inclure les VIs de `vi.lib` : inclut ou exclut les VIs de `vi.lib` dans la fenêtre Hiérarchie.



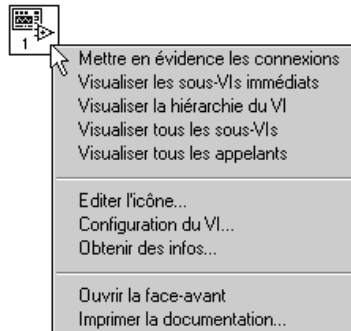
Inclure les variables globales : inclut ou exclut les variables globales dans la fenêtre Hiérarchie.



Inclure les définitions de type : inclut ou exclut les définitions de type dans la fenêtre Hiérarchie.

Menu local de nœud dans la fenêtre Hiérarchie

Si vous cliquez du bouton droit sur un nœud (sous-VI, variable globale ou définition de type) affiché dans la fenêtre Hiérarchie, un menu apparaît, offrant des commandes permettant de contrôler l’affichage ou d’effectuer des actions liées au nœud sélectionné.



Les commandes sont les suivantes :

- **Mettre en évidence les connexions** : fait du nœud sélectionné le nœud principal et souligne en rouge les bords reliant ses VIs appelants immédiats et ses sous-VIs.
- **Visualiser les sous-VIs immédiats** : si un nœud masque tous ou certains de ses sous-VIs, cet élément de menu développe le nœud pour afficher tous les sous-VIs immédiats. Les bords reliant le nœud à ses sous-VIs sont soulignés en rouge. L’élément de menu bascule avec **Masquer tous les sous-VIs**.
- **Masquer tous les sous-VIs** : si un nœud affiche tous ses sous-VIs immédiats, cet élément réduit le nœud pour masquer toute la chaîne de sous-VIs. L’élément de menu bascule avec **Visualiser les sous-VIs immédiats**.

- **Visualiser la hiérarchie des VIs** : fait du nœud sélectionné le nœud principal et affiche les nœuds appartenant à sa chaîne d'appel et chaîne de sous-VI. Les racines qui ne sont pas liées sont également visibles mais tous leurs sous-VIs restent masqués. Les bords connectant la chaîne d'appel d'un nœud et une chaîne de sous-VIs sont soulignés en rouge.
- **Visualiser tous les sous-VIs** : fait du nœud sélectionné le nœud principal et développe sa chaîne de sous-VI. Les bords connectant la chaîne de sous-VI du nœud sont soulignés en rouge.
- **Visualiser tous les VIs appelants** : fait du nœud sélectionné le nœud principal et développe sa chaîne d'appel. Les bords connectant la chaîne d'appel sont soulignés en rouge.
- **Editer l'icône...** : affiche l'éditeur d'icône pour permettre l'édition de l'icône du nœud.
- **Configuration du VI...** : affiche la boîte de dialogue Configuration du VI pour le nœud.
- **Obtenir des infos...** : affiche la boîte de dialogue Obtenir des infos pour le nœud.
- **Ouvrir la face-avant** : ouvre la face-avant du VI de la variable globale ou définition de type.
- **Imprimer la documentation...** : affiche la boîte de dialogue Imprimer la documentation, dans laquelle vous pouvez choisir la portion du VI à imprimer. Pour plus d'informations sur cette boîte de dialogue, consultez le section *Impression de la documentation* du chapitre 5, *Impression et documentation des VIs*.

Utilisation de la souris

La souris peut être utilisée pour la sélection des nœuds, la copie ou le déplacement de nœuds et pour des raccourcis de certains éléments de menu. Le cas échéant, ces séquences sont effectuées avec l'outil Flèche sélectionné.

Les raccourcis souris sont les suivants. Pour une description plus complète des éléments offerts dans le menu local, consultez la section *Menu local de nœud dans la fenêtre Hiérarchie* de ce chapitre.

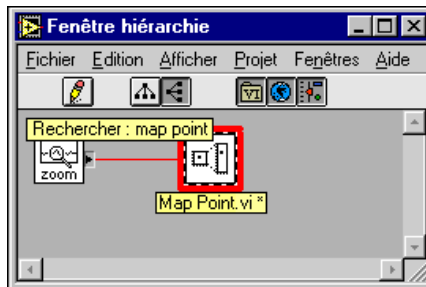
- Un clic sur le bouton en flèche rouge exécute l'action Visualiser les sous-VIs immédiats.
- <Maj>-clic sur le bouton en flèche rouge ou noire exécute l'action Visualiser tous les sous-VIs.

- <Ctrl-clic> (**Windows**), <option-clic> (**Macintosh**), <meta-clic> (**Sun**) et <Alt-clic> (**HP-UX**) sur le nœud exécute l'action Visualiser la hiérarchie des VIs.
- Un double-clic sur le nœud exécute l'action Ouvrir la face-avant.
- Un simple clic sur le nœud le sélectionne pour le faire glisser dans un diagramme ou le copier dans le Presse-papiers pour l'utiliser comme sous-VI.
- <Maj>-clic tout en maintenant le bouton de la souris enfoncé pendant la sélection permet de sélectionner plusieurs nœuds pour les copier dans d'autres diagrammes ou faces-avant. Vous pouvez également sélectionner plusieurs objets en plaçant un rectangle autour des objets à sélectionner.
- La touche <Tab> fait passer de l'outil Flèche à l'outil Main de la palette **Outils**.

Recherche de VIs dans la fenêtre Hiérarchie

Vous pouvez effectuer une recherche en tapant simplement le nom du nœud. Une petite fenêtre de recherche apparaît avec le texte que vous venez de taper. La recherche est immédiate et met en évidence le nœud correspondant en affichant l'info-bulle de son nom.

Dans l'illustration suivante, le mécanisme de recherche a trouvé le nœud `map_point.vi`.



Les recherches étant effectuées en même temps que la saisie, lorsqu'aucun nœud ne correspond aux caractères entrés dans la fenêtre de recherche, le système émet un bip et il devient impossible de saisir des caractères supplémentaires. Dans ce cas, utilisez la touche <Retour arrière> ou <Supprimer> pour effacer un ou plusieurs caractères et reprendre la saisie.

La fenêtre de recherche disparaît automatiquement si vous n'appuyez sur aucune touche pendant un certain temps. Vous pouvez appuyer sur la touche <Echap> pour immédiatement masquer la fenêtre de recherche.

Lorsqu'un nœud correspondant à la recherche est trouvé, vous pouvez utiliser la touche de direction droite ou bas, ou <Entrée> sous Windows et Sun ou <Retour> sur Macintosh et HP-UX, pour trouver le nœud suivant correspondant à la chaîne recherchée. Pour trouver le nœud correspondant précédent, appuyez sur la touche de direction gauche ou haut, ou <Maj-Entrée> sous Windows et Sun ou <Maj-Retour> sur Macintosh et HP-UX.

Recherche de VIs, d'objets et de texte

Lorsque vous développez une application comprenant plusieurs sous-VIs ou même un seul VI important, il se peut que vous souhaitiez rechercher des occurrences d'un objet ou d'une chaîne de texte particulier. L'option **Projet»Rechercher** peut vous aider à trouver toutes les instances des objets suivants avec les noms que vous indiquez :

- VIs
- Fonctions intégrées
- Définitions de type
- Variables globales
- Variables locales
- Attribute Nodes
- Points d'arrêt
- Terminaux de face-avant
- Texte

Pour plus d'informations sur les définitions de type, consultez la section *Définitions de type* du chapitre 24, *Commandes personnalisées et définitions de type*. Pour plus d'informations sur les variables locales et globales, consultez le chapitre 23, *Variables globales et locales*. Pour plus d'informations sur les attribute nodes, consultez le chapitre 22, *Attribute Nodes*. Pour plus d'informations sur les points d'arrêt, consultez la section *Mise en place de points d'arrêt* du chapitre 4, *Exécution et mise au point des VIs et sous-VIs*.

En plus de l'option **Rechercher**, plusieurs objets ont des éléments de menu local vous aidant à rapidement trouver des objets liés. Par exemple, si vous faites apparaître le menu déroulant d'une commande, un élément de menu vous aide à trouver le terminal correspondant ainsi que les variables locales ou les attributs nodes qui lui sont associés.

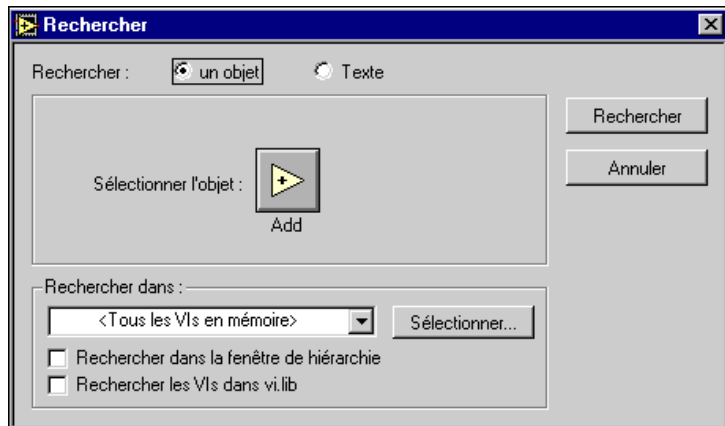
Boîte de dialogue Rechercher

Pour afficher la boîte de dialogue **Rechercher**, sélectionnez l'option **Projet>Rechercher...**, ou appuyez sur <Ctrl-f> (**Windows**), <commande-f> (**Macintosh**), <meta-f> (**Sun**) ou <Alt-f> (**HP-UX**)

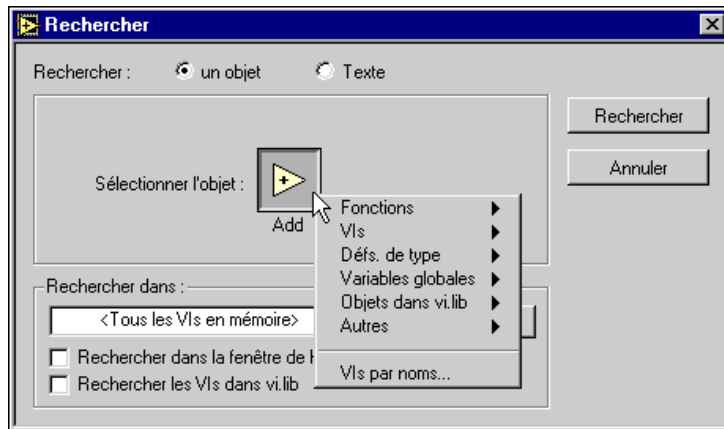
Un raccourci consiste à sélectionner un morceau de texte ou un objet avant d'accéder à la boîte de dialogue. La boîte de dialogue apparaît, avec le texte ou l'objet déjà entré pour la recherche.

Recherche de VIs et d'autres objets

Pour trouver des VIs ou d'autres objets, cliquez sur le bouton **Objets** dans la boîte de dialogue Rechercher.



Pour sélectionner l'objet à rechercher, cliquez sur le bouton placé après la chaîne **Sélectionner l'objet**. Le menu de Sélection d'objets apparaît.



Les éléments présents dans ce menu sont :

- **Fonctions** : affiche la palette **Fonctions** permettant de sélectionner des fonctions intégrées, des VIs ou d'autres objets personnalisés apparaissent dans la palette **Fonctions**.
- **VIs** : affiche une palette de tous les VIs en mémoire qui ne se trouvent pas dans `vi.lib`.
- **Défs. de type** : affiche une palette de toutes les définitions de type en mémoire qui ne se trouvent pas dans `vi.lib`.
- **Variables globales** : affiche une palette de tous les VIs globaux en mémoire qui ne se trouvent pas dans `vi.lib`.
- **Objets dans `vi.lib`** : affiche tous les VIs, définitions de type et variables globales qui se trouvent dans `vi.lib`.
- **Autres** : affiche un menu permettant de sélectionner des attributs, des points d'arrêt et des terminaux de face-avant.
- **VIs par noms...** : appelle la boîte de dialogue de sélection des VIs par nom, qui affiche tous les VIs en mémoire par nom et par ordre alphabétique. Vous pouvez taper le nom de l'objet pour rapidement passer à l'élément dans la liste souhaitée. Les trois cases (**VIs**, **Variables globales** et **Défs. de type**) vous permettent d'inclure ou d'exclure différents types d'objets à afficher dans la liste.

Recherche de texte

Pour rechercher des chaînes de caractères spécifiques, cliquez sur le bouton **Texte** dans la boîte de dialogue Rechercher.



Options de recherche

Vous pouvez taper le texte à rechercher et limiter l'ampleur de la recherche à certaines zones des VIs (face-avant, diagramme, info-bulles ou historique) ou à des composantes des objets, telles que l'étiquette ou la description.

Les éléments de menu suivantes permettent de préciser la recherche :

- **Respecter la casse** : trouve le texte dont la casse (majuscules ou minuscules) correspond exactement à celle des caractères entrés.
- **Respecter le mot entier** : trouve le texte uniquement s'il est précédé ou suivi d'un caractère non alphanumérique tel qu'un espace ou un signe plus (+), ou s'il se trouve au début ou à la fin d'une ligne (si cette option de menu n'est pas sélectionnée, n'importe quelle chaîne de caractères est proposée, qu'elle soit un fragment d'une chaîne plus longue ou non).
- **Expression courante** : interprète la chaîne de caractères comme une expression courante. L'expression courante a les mêmes spécifications que la fonction Rechercher une expression (voir **Aide en ligne**»**Référence sur les fonctions et VIs** pour ces spécifications).

Options de recherche de texte

Le bouton **Davantage d'options...** invoque la boîte de dialogue Options de recherche de texte, qui contient des éléments permettant de configurer l'étendue de la recherche de texte pour chaque VI.



Les éléments de menu indiquent si le texte visible ou masqué doit être cherché, et où le chercher. Ces éléments sont :

- **Recherche dans le VI** : indique si la recherche doit se faire dans la face-avant d'un VI, un diagramme, dans le texte de la boîte de dialogue Infos sur le VI ou dans le texte de l'historique. Une de ces cases doit obligatoirement être cochée.
- **Recherche dans l'étiquette de l'objet** : indique si la recherche doit se faire sur les étiquettes visibles ou masquées.
- **Recherche dans les données et composantes de l'objet** : indique si la recherche doit se faire sur des données et composantes visibles ou masquées. Les données et composantes d'un objet représentent tout ce qui appartient à un objet, à l'exception de son étiquette. Vous pouvez également effectuer une recherche dans la description des objets. Un des éléments de **Recherche dans l'étiquette de l'objet** et **Recherche dans les données et composantes de l'objet** doit obligatoirement être activé.



Remarque

Les options de recherche de texte sont enregistrées d'une recherche à l'autre. Si vous ne trouvez pas une chaîne qui est censée exister, vérifiez que les options de recherche sont bien définies.

Réduction de l'étendue de la recherche

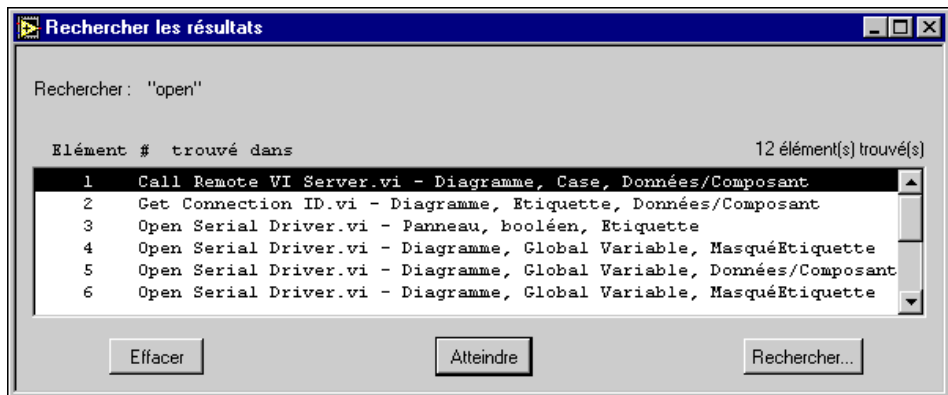
Trois éléments du menu déroulant **Rechercher dans** indiquent les VIs sur lesquels effectuer la recherche :

- **Tous les VIs en mémoire** : la recherche doit s'effectuer dans tous les VIs chargés en mémoire. Deux cases permettent d'inclure ou d'exclure les VIs de `vi.lib` et de la fenêtre Hiérarchie.
- **VIs sélectionnés** : sélectionne un jeu personnalisé de VIs à rechercher. La fenêtre Hiérarchie peut être incluse ou exclue de la zone de recherche. Le bouton **Sélectionner...** affiche l'élément de menu **Sélectionner les VIs** dans la boîte de dialogue Rechercher.
- **Nom d'un VI** : recherche un VI spécifique uniquement. Le nom du VI dans cet élément de menu déroulant dépend du VI actif au moment de l'invocation de la boîte de dialogue Rechercher.

Cliquez sur le bouton **Rechercher** pour commencer la recherche pour le texte sélectionné.

Fenêtre Résultat de la recherche

Une fois la recherche terminée et s'il existe plus d'un résultat, la fenêtre Résultat de la recherche affiche tous les résultats, tel que présenté dans l'illustration suivante. Si un seul résultat est trouvé, il est immédiatement mis en surbrillance, sans que la fenêtre Résultat de la recherche ne soit affichée.



Vous pouvez afficher cette fenêtre en sélectionnant **Projet>Résultat de la recherche...**

Les éléments de menu suivants sont utilisables dans la fenêtre Résultat de la recherche :

- **Effacer** : efface et libère la mémoire utilisée pour le stockage des résultats de la recherche.
- **Atteindre** : met le résultat sélectionné en surbrillance. Vous pouvez également parvenir en double-cliquant sur l'élément. Les éléments déjà mis en surbrillance sont identifiés par une encoche.
- **Rechercher...** : appelle la boîte de dialogue Rechercher.
- **Stop** : apparaît à la place du bouton **Rechercher...** pendant la recherche. Pour une longue recherche dans de nombreux VIs, la fenêtre Résultat de la recherche est mise à jour à chaque fois qu'un résultat est trouvé et contient une indication de l'état de la recherche. Vous pouvez appuyer sur le bouton **Stop** pour arrêter la recherche.

Si vous supprimez un objet ou un VI qui fait partie du résultat de la recherche, l'entrée de ce résultat est désactivée. Remarquez que la modification du texte ne met pas les résultats à jour. Il se peut donc que vous mettiez en surbrillance un résultat de recherche ne correspondant plus aux critères. Le cas échéant, effectuez une autre recherche pour mettre les résultats à jour.

Recherche d'éléments de recherche précédents ou suivants

Sélectionnez l'option **Projet»Suivant** pour mettre en surbrillance les résultats précédents et suivants dans la liste des résultats. Pour le texte masqué, tous les objets nécessaires à l'affichage du texte sont rendus temporairement visibles sous la forme d'objets grisés avec le texte sélectionné. Lorsque vous cliquez avec la souris ou appuyez sur une touche, les objets sont à nouveau masqués.

Menu local de recherche pour les variables globales et locales et les Attribute Nodes

Un menu local contenant une option de recherche est utilisable sur les commandes et indicateurs de la face-avant, les variables globales et locales et les attribute nodes. Vous pouvez faire apparaître un menu local sur une variable locale ou un attribute node pour trouver sa commande, son terminal de face-avant ou d'autres variables locales et attribute nodes. Les commandes et indicateurs de face-avant disposent de menus locaux permettant de rechercher toutes les variables locales et attribute nodes correspondants. Depuis un nœud de variable globale, vous pouvez trouver la définition correspondante ou tous les autres nœuds de variable globale. Les commandes de variables globales disposent de menus locaux permettant de rechercher toutes les références correspondantes en mémoire. Si plusieurs résultats sont trouvés, la fenêtre Résultat de la recherche apparaît.

Exécution et mise au point des VIs et sous-VIs

Ce chapitre décrit le fonctionnement et la mise au point des VIs, ainsi que la configuration des VIs et des sous-VIs pour des modes d'exécution spéciaux.

Exécution des VIs

Exécution des VIs



Bouton Exécuter

Vous pouvez exécuter un VI en sélectionnant **Exécution»Exécuter** ou en cliquant sur le bouton Exécuter. Si nécessaire, le G compile le VI.



VI s'exécutant
au niveau principal

Lorsque le VI s'exécute à son niveau principal (ce qui signifie qu'il n'a pas de VI appelant et qu'il n'est donc pas un sous-VI), le bouton **Exécuter** se transforme de la manière ci-contre.



VI appelant
en cours d'exécution

Si le VI s'exécute en tant que sous-VI, le bouton **Exécuter** se transforme de la manière ci-contre.



Bouton Exécuter
en continu

Appuyez sur le bouton **Exécuter en continu** pour exécuter indéfiniment le VI, jusqu'à ce que vous demandiez un abandon ou une pause de l'exécution.



Bouton Abandonner

En appuyant sur le bouton **Abandonner**, vous abandonnez l'exécution du VI *principal*. Si un VI est utilisé par plusieurs VIs principaux en cours d'exécution, le bouton apparaît grisé.



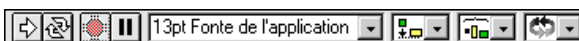
Bouton Pause

En appuyant sur ce bouton, vous suspendez l'exécution. Vous pouvez appuyer à nouveau dessus à n'importe quel moment pour continuer l'exécution.

Vous pouvez exécuter plusieurs VIs en même temps. Une fois que vous avez lancé le premier, passez à la face-avant ou au diagramme du suivant et lancez-le comme cela a été préalablement décrit. Notez que si vous exécutez un sous-VI en tant que VI principal, tous les VIs appelants sont suspendus jusqu'à la fin de l'exécution du sous-VI. Vous ne pouvez pas exécuter un sous-VI en même temps en tant que VI principal et en tant que sous-VI.

Si votre VI s'exécute mais ne fonctionne pas correctement, consultez les sections *Réparation des VIs invalides* et *Mise au point des VIs exécutables* plus loin dans ce chapitre.

Lorsque vous éditez un VI, la barre d'outils contient plusieurs outils utilisés pour l'édition des VIs, dont le menu Police de caractères, le menu Alignement, le menu Distribution et le menu Réorganiser.

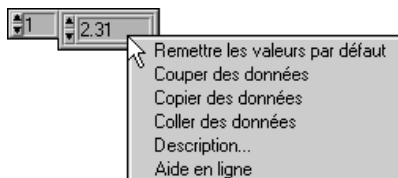


Lorsque vous exécutez un VI, ces outils sont remplacés par des outils de mise au point, comme le montre l'illustration suivante.



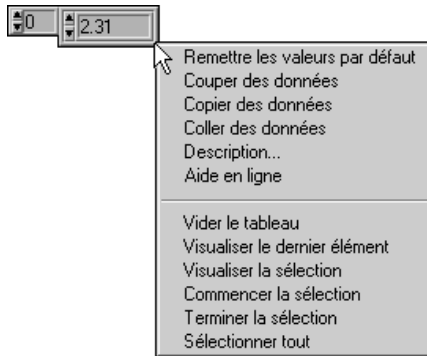
Bouton Exécution

Si vous ouvrez le menu local d'une commande pendant l'exécution d'un VI, vous pouvez constater que tous les objets ont un ensemble beaucoup plus simple d'éléments de menu. Ceux-ci sont les mêmes que ceux du sous-menu **Opérations sur les données** que vous voyez dans les menus locaux lorsque vous éditez des VIs.

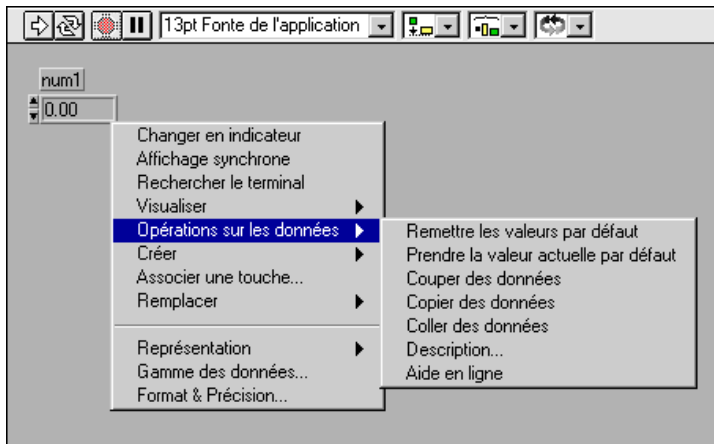


Vous pouvez couper, copier ou coller le contenu de la commande, régler la commande sur sa valeur par défaut et lire la description de la commande à l'aide des éléments figurant dans ce menu. Certaines commandes plus

complexes ont des options supplémentaires. Par exemple, un tableau a des éléments de menu pour copier une gamme de valeurs ou pour atteindre le dernier élément du tableau.



Lorsque vous éditez un VI, le menu local d'un objet contient un menu d'édition et un sous-menu **Opérations sur les données**, comme le montre l'illustration suivante. Ce sous-menu contient les mêmes éléments que vous voyez lorsque vous exécutez un VI plus un élément supplémentaire, **Prendre la valeur actuelle par défaut**.



Le menu **Exécution** sur la barre de menus contient des options pour exécuter le VI actuel. Les sections suivantes abordent plusieurs tâches fondamentales associées à l'exécution.

Arrêt des VIs



Bouton Abandonner

Dans des circonstances ordinaires, vous laissez l'exécution d'un VI se poursuivre jusqu'à ce qu'elle soit terminée. Cependant, si vous avez besoin d'arrêter immédiatement l'exécution, cliquez sur le bouton **Abandonner** ou sélectionnez **Exécuter»Stop**. Le bouton **Abandonner** arrête le VI principal à la première occasion. Le VI arrêté n'a probablement pas terminé sa tâche, donc vous ne pouvez pas vous fier aux données qu'il produit. Bien que le langage G ferme les fichiers ouverts à ce moment et interrompe toute acquisition de données pouvant être en cours, évitez de construire des VIs qui ont recours au bouton **Abandonner** ou à l'élément de menu **Stop**. Par exemple, si vous créez un VI qui s'exécute indéfiniment au sein d'une boucle While jusqu'à ce qu'il soit arrêté par l'opérateur, câblez le terminal conditionnel de la boucle à un interrupteur booléen de la face-avant.

Si vous voulez éviter qu'un opérateur n'interrompe votre VI par inadvertance en cliquant sur le bouton **Abandonner**, cachez-le en désélectionnant la case **Configuration du VI»Options des fenêtres» Visualiser le bouton Abandonner** à partir du menu local du cadre icône de la face-avant du VI. Pour plus d'informations, consultez la section *Options des fenêtres* du chapitre 6, *Configuration des VIs et sous-VIs*.

Exécution répétée des VIs



Bouton Exécution permanente

Pour exécuter un VI de façon répétée, cliquez sur le bouton **Exécution permanente**. L'exécution du VI commence immédiatement, et les flèches vides du bouton **Exécution permanente** se transforment en flèches pleines pendant l'exécution du VI. Cliquez à nouveau sur le bouton **Exécution permanente** pour arrêter le VI. Le VI s'arrête lorsqu'il arrive normalement à son terme.



VI en cours d'exécution continue

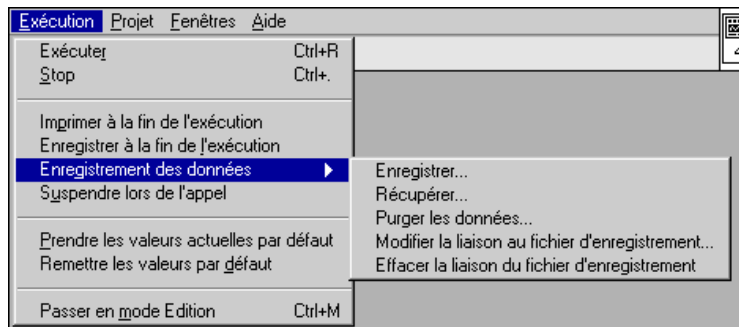
Le comportement du VI et l'état de la barre d'outils pendant une exécution permanente sont les mêmes qu'au cours d'une exécution unique lancée avec le bouton **Exécuter** ou la commande **Exécution»Exécuter**.

Enregistrement des données de la face-avant

L'opération d'enregistrement des données de la face-avant écrit un horodatage et les données de toutes les commandes de la face-avant d'un VI dans un fichier d'enregistrement de données. Vous pouvez avoir plusieurs fichiers différents, contenant chacun des données de différents tests. Vous pouvez accéder à ces données en employant une récupération interactive à partir du VI, une récupération de données contrôlée par programme ou des fonctions standard d'E/S de fichier.

Chaque VI conserve une liaison du fichier d'enregistrement, représentant l'emplacement du fichier d'enregistrement de données dans lequel sont enregistrées les données de la face-avant. Si la liaison est vierge lorsque l'enregistrement des données est activé, ce qui signifie qu'aucun emplacement n'est spécifié, le VI vous demande de préciser l'emplacement du fichier.

Configurez et contrôlez l'enregistrement des données de la face-avant en sélectionnant le menu **Exécuter** et son sous-menu **Enregistrement des données**, comme le montre l'illustration suivante.



Lorsque l'enregistrement automatique des données est activé pour un VI, le langage G enregistre la face-avant du VI à la fin de chaque exécution du VI. Vous pouvez savoir si l'enregistrement automatique des données est activé pour un VI en regardant l'élément de menu **Exécution»Enregistrer à la fin de l'exécution**. En sélectionnant **Exécution»Enregistrer à la fin de l'exécution**, vous activez l'enregistrement automatique des données pour un VI si celui-ci était désactivé, et vous désactivez l'enregistrement automatique des données si celui-ci était activé.

Pour enregistrer de façon interactive les données de la face-avant d'un VI, sélectionnez **Exécution»Enregistrement des données»Enregistrer...**



Remarque

Un graphe déroulant ne retourne qu'un point à la fois lors de l'enregistrement de données de la face-avant. Si vous câblez un tableau sur l'indicateur d'un graphe déroulant, le fichier d'enregistrement des données contient le sous-ensemble du tableau affiché sur le graphe pour cet enregistrement.

Vous pouvez changer la liaison du fichier d'enregistrement d'un VI avec l'élément de menu **Exécution»Enregistrement des données»Changer la liaison du fichier d'enregistrement...**



Bouton Entrée

Vous pouvez sélectionner un enregistrement spécifique en tapant le numéro de l'enregistrement et en cliquant ensuite sur le bouton **Entrée**, ou en appuyant sur la touche <Entrée> du pavé numérique.



Bouton Effacer

Vous pouvez marquer l'enregistrement sélectionné pour qu'il soit effacé en cliquant sur le bouton **Effacer**. Lorsque l'enregistrement sélectionné est marqué pour être effacé, le bouton **Effacer** se transforme en une poubelle pleine. Vous pouvez désélectionner l'enregistrement en cliquant sur la poubelle pleine. Les enregistrements sélectionnés ne sont pas effacés tant que vous ne sélectionnez pas **Exécution»Enregistrement des données»Purger les données...**, ou tant que vous ne quittez pas le mode de récupération de données, soit en cliquant sur le bouton **OK**, soit en sélectionnant à nouveau **Récupérer...** S'il reste des enregistrements marqués pour être effacés lorsque vous quittez le mode de récupération de données, le système vous demande si vous voulez effacer les enregistrements marqués.

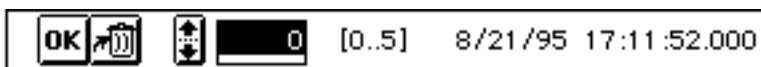


Bouton Enregistrement marqué pour l'effacement

Cliquez sur le bouton **OK** pour revenir au VI auquel correspond le fichier d'enregistrement de données que vous consultez.

En sélectionnant **Exécution»Effacer la liaison du fichier d'enregistrement**, vous dissociez le VI de tout fichier d'enregistrement associé. La prochaine fois que vous procéderez à un enregistrement de données à partir de la face-avant du VI, le système vous demandera de préciser un fichier d'enregistrement des données.

Pour consulter les données enregistrées de manière interactive, sélectionnez l'élément de menu **Exécution»Récupérer...** La barre d'outils se transforme en barre d'outils de récupération de données, comme le montre l'illustration suivante.



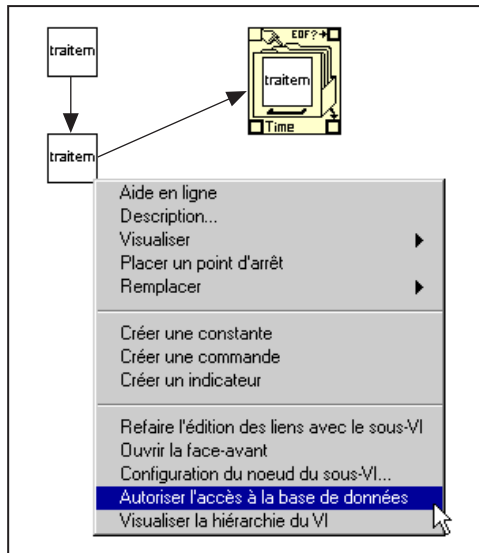
Le nombre qui apparaît en surbrillance indique l'enregistrement qui est actuellement sélectionné pour être affiché. Les nombres entre crochets indiquent la gamme d'enregistrements accessibles. La date et l'heure indiquent le moment auquel l'enregistrement sélectionné a été effectué. Vous pouvez afficher l'enregistrement suivant ou précédent en cliquant respectivement sur la flèche de déplacement vers le haut ou sur la flèche de déplacement vers le bas. Vous pouvez également utiliser les flèches de déplacement vers le haut ou vers le bas qui se trouvent sur votre clavier.

Récupération de données contrôlée par programme

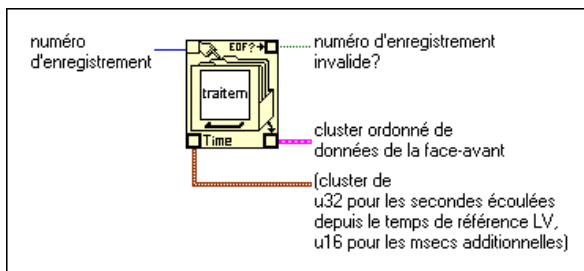
Vous pouvez récupérer des données enregistrées à partir d'un VI en utilisant l'élément de menu **Autoriser l'accès à la base de données** ou en utilisant des fonctions standard d'E/S de fichier pour lire le fichier d'enregistrement de données.

Accès aux bases de données

Vous pouvez récupérer des données en utilisant l'élément de menu **Autoriser l'accès à la base de données**. Vous pouvez y accéder à partir du menu local d'un sous-VI qui a enregistré des données de face-avant dans le fichier d'enregistrement de données associé au VI, comme le montre l'illustration suivante. Un *halo* semblable à une armoire de classement apparaît autour du fichier d'enregistrement de données. Ce halo a des terminaux pour accéder aux données du fichier d'enregistrement de données. Si vous exécutez le diagramme appelant, le sous-VI ne s'exécute pas. A la place, vous pouvez récupérer des données d'un enregistrement spécifié du fichier d'enregistrement de données. Il vous donne également l'heure à laquelle les données ont été enregistrées et une valeur booléenne indiquant si le numéro d'enregistrement indiqué est incorrect.



L'illustration suivante montre les terminaux du halo pour accéder aux données enregistrées.



Si le sous-VI a n enregistrements effectués, vous pouvez transmettre n'importe quel nombre de $-n$ à $n-1$ au terminal de numéro d'enregistrement. Vous pouvez accéder aux données relatives au *premier* enregistrement effectué en utilisant des numéros d'enregistrement non négatifs. 0 désigne le premier enregistrement, 1 désigne le deuxième enregistrement, et ainsi de suite, jusqu'à $n-1$, qui correspond au dernier enregistrement.

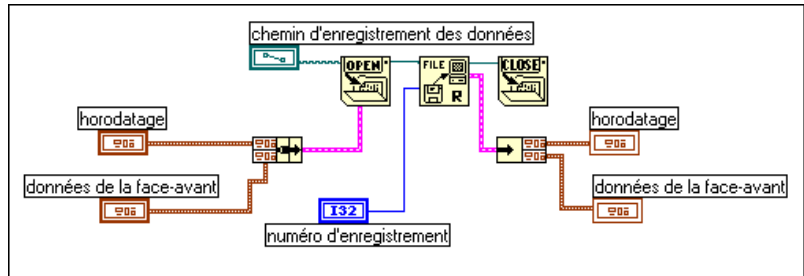
Vous pouvez accéder aux données relatives au *dernier* enregistrement effectué en utilisant des numéros d'enregistrement négatifs. N'oubliez pas que -1 désigne le dernier enregistrement, -2 désigne l'avant-dernier, et ainsi de suite, jusqu'à $-n$, qui correspond au premier enregistrement. Si vous transmettez un nombre en dehors de la gamme $-n$ à $n-1$ au terminal de numéro d'enregistrement, la sortie "numéro d'enregistrement incorrect ?" est fixée sur Vrai (TRUE), et aucune donnée n'est récupérée.

Récupération de données à l'aide de fonctions d'E/S de fichier

Vous pouvez également récupérer des données enregistrées à partir de la face-avant d'un VI en utilisant les fonctions d'E/S de fichier du G. Chaque enregistrement du fichier créé lors de l'enregistrement des données de la face-avant est un cluster contenant un horodatage et les données de la face-avant. Le cluster d'horodatage consiste en un entier non signé 32 bits, représentant les secondes, et en un entier non signé 16 bits, représentant les millisecondes écoulées depuis l'instant de référence du G. Ce cluster est suivi d'un cluster des données de la face-avant dans l'ordre de la face-avant.

Vous pouvez accéder aux données des fichiers créés lors de l'enregistrement des données de la face-avant en utilisant les mêmes fonctions d'E/S de fichier du G que celles que vous utilisez pour accéder

aux fichiers d'enregistrement de données créés par un programme. Entrez le type approprié, décrit préalablement, comme type désiré pour la fonction Ouvrir un fichier, comme cela est illustré dans l'exemple suivant.



Mise au point des VIs

Cette section décrit toute une gamme de solutions pour corriger les erreurs et résoudre les problèmes pouvant survenir avec les instruments virtuels.

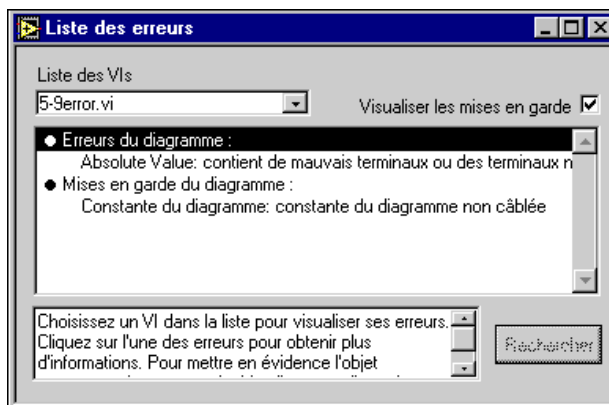
Réparation des VIs invalides

Un VI ne peut être ni compilé, ni exécuté s'il est invalide. Le bouton **Exécution du VI** est généralement brisé tant que vous créez ou que vous éditez le VI, jusqu'à ce que vous ayez câblé toutes les icônes du diagramme. S'il est toujours brisé une fois que vous avez fini de câbler les icônes, essayez de sélectionner **Edition»Supprimer les fils incorrects** ou de sélectionner <Ctrl-b> (**Windows**), <commande-b> (**Macintosh**), <meta-b> (**Sun**) ou <Alt-b> (**HP-UX**). Souvent, cela suffit pour réparer un VI invalide.



Bouton
Flèche Brisée

Pour déterminer pourquoi un VI est invalide, cliquez sur le bouton **Flèche Brisée**. La fenêtre Liste des erreurs apparaît, indiquant toutes les erreurs. Cette fenêtre est représentée dans l'illustration suivante.



Bouton
Mise en garde

Vous pouvez également accéder à cette boîte en cliquant sur le bouton **Mise en garde** d'un VI, ou en sélectionnant **Fenêtres»Liste des erreurs**. Le bouton **Mise en garde** correspondant à un VI est visible uniquement si le VI présente une mise en garde (par exemple, des objets qui se chevauchent) et si vous cochez **Visualiser les mises en garde** dans la fenêtre Liste des erreurs. Utilisez la boîte de dialogue **Préférences** pour configurer le G de manière à afficher par défaut les mises en garde.

Pour localiser une erreur, double-cliquez sur le texte qui la décrit. Lors de l'affichage d'une erreur, la fenêtre concernée apparaît au premier plan et l'objet causant l'erreur est mis en surbrillance. Visualisez les erreurs et les mises en garde pour d'autres VIs en les sélectionnant dans le menu local **Liste des VIs**.

La liste suivante contient certaines des raisons les plus courantes pour lesquelles un VI peut être invalide après avoir été édité.

- Un terminal de fonction exigeant une entrée n'est pas câblé. Par exemple, vous devez câbler toutes les entrées des fonctions arithmétiques. Vous ne pouvez pas laisser de fonctions non câblées sur le diagramme pendant que vous exécutez le VI pour essayer différentes configurations.
- Le diagramme contient un fil de liaison brisé à cause d'une mauvaise concordance du type de données ou parce qu'une extrémité est déconnectée. Enlevez les fils de liaison brisés, y compris les morceaux de fil de liaison que vous ne pouvez pas voir, avec la commande **Edition»Supprimer les fils incorrects**.

- Un sous-VI est invalide, ou vous avez édité son connecteur après avoir placé son icône dans le diagramme.
- Le problème peut être causé par un objet rendu invisible, désactivé ou modifié d'une manière quelconque par un attribut node. Si possible, restaurez l'objet à l'aide de l'attribut node pour résoudre le problème.

Interprétation des messages d'erreur

La liste suivante contient certaines erreurs possibles, sous une forme similaire à celle sous laquelle elles se présentent dans la fenêtre Liste des erreurs.

Tableau 4-1. Messages d'erreur

Message d'erreur	Description
Nom de la fonction : contient un terminal non câblé ou incorrect.	Une entrée requise n'est pas câblée ou elle est d'un type incorrect. Câblez les entrées requises avec un type de données correct.
Code Interface Node : code objet non chargé.	Vous n'avez pas effectué correctement la liaison du code objet d'un CIN. Ouvrez le menu local du nœud et rechargez le code.
Commande : la commande ne correspond pas à sa définition de type.	Éditez la commande pour qu'elle corresponde, ou ouvrez le menu local et mettez à jour ou déconnectez le VI de définition de type de la commande.
L'énumération contient des entrées en double.	Tous les éléments d'une commande de type énumération doivent être uniques et distincts.
Davantage d'erreurs...	La fenêtre des erreurs affiche seulement 100 erreurs à la fois. Les erreurs qui ne sont pas listées apparaissent une fois que vous avez corrigé les erreurs listées.
Boucle For : N n'est pas câblé et il n'y a pas de tunnel d'indexation d'entrée.	A moins que N ne soit câblé ou qu'un tableau ne soit indexé en entrée, la boucle ne peut pas déterminer combien d'itérations elle doit exécuter.
Variable globale ou locale : la composante indiquée n'existe pas.	Le nom d'une variable a changé depuis la dernière fois que vous avez chargé le VI et ne correspond plus au nom qui figure dans la variable globale ou locale. Ouvrez le menu local de la variable et sélectionnez un autre nom.

Tableau 4-1. Messages d'erreur (Suite)

Message d'erreur	Description
Variable globale : sous-VI manquant.	Lors du chargement du VI appelant, le sous-VI global n'a pas été trouvé, peut-être parce que vous avez changé son nom. Remplacez le sous-VI global incorrect par un sous-VI correct.
Nœud : un VI à priorité de type sous-programme ne peut pas contenir un nœud asynchrone.	Vous ne pouvez pas utiliser une fonction asynchrone, telle que Attendre, dans un VI dont le niveau de priorité est de type sous-programme.
Registre à décalage droit : le type n'est pas défini.	Vous devez enlever les registres à décalage inutilisés.
Registre à décalage droit : certains des terminaux gauches, mais pas tous, sont câblés.	Un registre à décalage doit avoir des entrées pour tous les côtés gauches ou pour aucun.
Variable locale de séquence : une ou plusieurs des variables locales de séquence n'ont jamais été assignées.	Vous avez oublié de câbler vers une variable locale de structure Séquence. Enlevez les variables locales de séquence inutilisées.
Nom de sous-VI : mauvaise liaison au sous-VI.	La disposition des connecteurs du sous-VI a changé depuis la dernière fois que vous avez chargé le sous-VI, ou vous avez oublié d'assigner des commandes ou indicateurs du sous-VI à des terminaux de connecteur. Dans le premier cas, utilisez la commande Refaire l'édition des liens dans le menu local du sous-VI.
Nom de sous-VI : références récursives (éliminez-les).	Vous avez changé le nom du VI appelant pour lui donner le nom de l'un de ses sous-VIs, et maintenant le G interprète ce changement comme un appel récursif du VI à lui-même (le VI et le sous-VI sont probablement dans des répertoires différents). Le G empêche la récursion lorsque vous essayez de placer un VI sur son propre diagramme.
Nom de sous-VI : erreur de liaison avec un sous-programme de LabVIEW.	Il y a un problème au niveau de la liaison à un programme CIN.
Nom de sous-VI : un VI à priorité de type sous-programme ne peut pas appeler un sous-VI qui n'a pas une priorité de type sous-programme.	Vous devez faire en sorte que le sous-VI s'exécute avec une priorité de type sous-programme ou changer le VI appelant pour que sa priorité soit autre qu'une priorité de type sous-programme.

Tableau 4-1. Messages d'erreur (Suite)

Message d'erreur	Description
Nom de sous-VI : le sous-VI est déjà en cours d'exécution.	Vous ne pouvez pas exécuter un VI si l'un de ses sous-VIs est déjà en cours d'exécution comme sous-VI d'un autre VI.
Nom de sous-VI : le sous-VI est soit en mode d'ordonnancement de la face-avant, soit en mode d'ordonnancement du cluster.	Vous ne pouvez pas exécuter un VI si vous changez l'ordonnancement de la face-avant ou du cluster de l'un de ses sous-VIs.
Nom de sous-VI : le sous-VI est en mode de récupération interactive.	Vous ne pouvez pas exécuter un VI si l'un de ses sous-VIs est en mode de récupération interactive.
Nom de sous-VI : sous-VI manquant.	Lors du chargement du VI appelant, le sous-VI n'a pas été trouvé, peut-être parce que vous avez changé son nom. Remplacez le sous-VI incorrect par un sous-VI correct, ou ouvrez le sous-VI manquant si vous le trouvez.
Nom de sous-VI : le sous-VI n'est pas exécutable.	Le sous-VI est invalide. Ouvrez-le et réparez ses erreurs.
Terminal : le tableau ou le cluster associé sur la face-avant n'a aucun élément ; son type n'est pas défini.	Vous devez mettre une commande ou un indicateur dans le tableau ou dans le cluster.
Définition de type : impossible de trouver une définition de type valide.	Lors du chargement du VI appelant, le VI de définition de type n'a pas été trouvé, peut-être parce que vous avez changé son nom. Ouvrez le VI de définition de type manquant si vous le trouvez, ouvrez le menu local de la commande incorrecte, ou remplacez-la par une commande correcte.
(Dés)Assembler par nom : cluster vide, ou certains composants n'ont pas de nom.	Le cluster d'entrée câblé à la fonction "Assembler par nom" ou "Désassembler par nom" est vide, ou il contient des composants qui ne portent pas d'étiquette.
Unité : mauvaise syntaxe d'unité.	Le texte dans le nœud n'est pas une expression d'unité légale. Un ? est placé immédiatement avant le caractère inconnu.

Si vous rencontrez des messages dont le sens n'est pas évident et qui ne sont pas expliqués dans ce chapitre, contactez National Instruments.

Mise au point des VIs exécutable

Si votre programme s'exécute mais ne produit pas les résultats attendus, vous pouvez souvent résoudre le problème en adoptant les mesures suivantes.

- Éliminez toutes les mises en garde du VI. Si vous sélectionnez **Visualiser les mises en garde** dans la fenêtre Liste des erreurs, la boîte-liste identifie les mises en garde pour votre VI. Déterminez leurs causes et éliminez-les.
- Vérifiez les chemins de câblage pour vous assurer que les fils de liaison sont connectés aux bons terminaux. En triple-cliquant sur le fil de liaison avec l'outil Doigt, vous pouvez mettre en surbrillance la totalité du chemin. Il est possible qu'un fil de liaison qui semble émaner d'un terminal émane d'un autre. Regardez attentivement pour voir l'endroit où l'extrémité du fil de liaison est connectée au nœud.
- Utilisez la fenêtre Aide (**Aide**»**Visualiser l'aide**) pour vous assurer que les fonctions sont correctement câblées.
- Vérifiez que la valeur par défaut des entrées non câblées des fonctions ou des sous-VIs est celle que vous escomptez.
- Utilisez des points d'arrêt, des exécutions en mode Animation et des exécutions en mode pas à pas pour déterminer si le VI s'exécute comme prévu.
- Utilisez la fonction de sonde décrite dans la section *Utilisation de l'outil Sonde* de ce chapitre pour observer les valeurs intermédiaires de certaines données. De plus, vérifiez la sortie d'erreur des fonctions et des sous-VIs, en particulier ceux qui effectuent des E/S.
- Observez le comportement du VI ou du sous-VI avec diverses valeurs d'entrée. Pour les commandes numériques à virgule flottante, entrez les valeurs NaN et $\pm\text{Inf}$ en plus des valeurs normales.
- Si l'exécution de votre VI est plus lente que prévu, vérifiez que l'exécution en mode Animation est désactivée dans les sous-VIs. De plus, fermez les fenêtres des sous-VIs lorsque vous ne les utilisez pas.
- Vérifiez la représentation de vos commandes et indicateurs pour voir si vous recevez un débordement parce que vous avez converti un nombre à virgule flottante en entier ou un entier en entier plus petit. Reportez-vous également à la section de ce chapitre intitulée *Reconnaître des données non définies*.
- Vérifiez la gamme des données des commandes et indicateurs et leur réaction face aux erreurs de dépassement de gamme. Ils peuvent ne pas réagir comme vous le voulez en cas d'erreur.

- Déterminez si des boucles For peuvent exécuter par inadvertance zéro itération et produire des tableaux vides. De plus, reportez-vous aux sections de ce chapitre intitulées *Exécution en mode Animation* et *Reconnaître des données non définies*.
- Vérifiez que vous avez correctement initialisé les registres à décalage, à moins que vous ne vouliez spécifiquement y sauvegarder des données d'une exécution de la boucle à la suivante.
- Vérifiez l'ordonnement des éléments du cluster au point source et au point de destination. Bien que le G détecte les défauts de concordance de type de données et de taille de cluster au moment de l'édition, le G ne détecte pas les défauts de concordance du type des éléments. Utilisez l'élément de menu **Ordonner le cluster...** dans le menu local du squelette du cluster pour vérifier l'ordonnement du cluster.
- Vérifiez l'ordre d'exécution des nœuds. Les nœuds qui ne sont pas connectés par un fil de liaison peuvent s'exécuter dans n'importe quel ordre. La disposition de ces nœuds dans l'espace ne contrôle pas leur ordre d'exécution. C'est-à-dire, les nœuds non connectés ne s'exécutent *pas* de gauche à droite et de haut en bas sur le diagramme, comme c'est le cas pour les instructions des langages textuels.
- Vérifiez qu'il n'y a pas de VI étranger. A l'inverse des fonctions, les sous-VIs non câblés ne génèrent pas toujours des erreurs (à moins que vous ne configuriez une entrée comme nécessaire ou recommandée). Si vous placez par erreur un sous-VI non câblé sur le diagramme, il s'exécute en même temps que le diagramme et, par conséquent, votre programme peut, de ce fait, effectuer des opérations superflues.
- Vérifiez qu'il n'y a pas de VI masqué. Vous pouvez, par inadvertance, avoir masqué des sous-VIs de trois manières : en en plaçant un directement au-dessus d'un autre nœud, en réduisant la taille d'une structure sans garder le sous-VI en vue ou en en plaçant un en dehors de la zone principale du diagramme. Dans le dernier cas, faites défiler la fenêtre du diagramme jusqu'à ses limites. En outre, comparez l'inventaire des sous-VIs utilisés dans le VI avec les éléments des sous-menus **Sous-VIs appelés par ce VI** et **Sous-VIs non ouverts**, dans le menu **Projet**, afin de déterminer s'il y a un quelconque sous-VI étranger. De plus, regardez dans la fenêtre Hiérarchie pour voir les sous-VIs correspondant à un VI, ou dans la fenêtre Liste des erreurs pour voir une liste des objets cachés (à condition que la case **Visualiser les mises en garde** soit cochée). Pour vous aider à éviter tout résultat incorrect causé par des VIs cachés, vous pouvez demander que les entrées des VIs soient nécessaires ; pour plus d'informations,

consultez la section *Connexions nécessaires, recommandées et optionnelles pour les sous-VIs* du chapitre 3, *Utilisation des sous-VIs*.

Comprendre les mises en garde

Si un objet est complètement caché par un autre objet, le G génère une mise en garde pour indiquer que tous les objets ne sont pas visibles. Par exemple, si un terminal est caché sous le bord d'une structure, ou si des tunnels sont les uns au-dessus des autres, un message de mise en garde est placé dans la fenêtre Liste des erreurs. Les mises en garde ne vous empêchent pas d'exécuter un VI ; elles sont destinées à vous aider à identifier des problèmes potentiels dans vos programmes.

Reconnaître des données non définies

Il y a deux valeurs symboliques qui peuvent apparaître dans les affichages numériques à virgule flottante pour indiquer des calculs erronés ou des résultats dépourvus de sens. *NaN* (de l'anglais "not a number" [pas un nombre]) est le symbole représentant une valeur à virgule flottante particulière que certaines opérations, telles que l'extraction de la racine carrée d'un nombre négatif, peuvent produire. *Inf* est une autre valeur à virgule flottante spéciale qui est produite, par exemple, en divisant un par zéro.

Des données non définies peuvent corrompre toutes les opérations ultérieures. Les opérations à virgule flottante propagent *NaN* et \pm *Inf*, qui, une fois convertis explicitement ou implicitement en entiers ou booléens, deviennent des valeurs aberrantes. Par exemple, la division de un par zéro produit *Inf*, mais la conversion de cette valeur en entier mot produit la valeur 32767, qui semble être une valeur normale. Avant de les convertir en entiers, vérifiez les valeurs à virgule flottante intermédiaire pour confirmer leur validité, à moins que vous ne soyez sûr que ce type d'erreur ne se produit pas dans votre VI.

Exécuter une boucle For zéro fois peut produire des valeurs inattendues. Par exemple, la sortie d'un registre à décalage initialisé est la valeur initiale. Cependant, si vous n'initialisez pas le registre à décalage, la sortie est soit la valeur par défaut pour le type de données (0, FALSE ou une chaîne de caractères vide), soit la dernière valeur chargée dans le registre à décalage lors de la dernière exécution du diagramme.

Une indexation au-delà de la dimension d'un tableau produit la valeur par défaut pour le type de données de l'élément du tableau. Vous pouvez faire cela par inadvertance de plusieurs manières, par exemple en indexant un tableau au-delà du dernier élément à l'aide d'une boucle While, en

fournissant une valeur trop élevée à l’index donné en entrée de la fonction “Indexer un tableau”, ou en fournissant un tableau vide à la fonction “Indexer un tableau”.

Lorsque vous élaboriez des VIs qui peuvent produire des valeurs de sortie non définies pour certaines valeurs d’entrée, ne vous fiez pas à des valeurs spéciales, telles que NaN, ni à des tableaux vides pour identifier le problème. Veillez plutôt à ce que votre VI produise un message d’erreur qui identifie le problème, ou à ce qu’il produise seulement des données par défaut définies.

Par exemple, si vous créez un VI qui utilise un tableau en amont pour auto-indexer une boucle For, évaluez le comportement du VI si le tableau d’entrée est vide. Générez un code d’erreur de sortie ou remplacez les valeurs créées par la boucle par des valeurs prédéfinies.

 **Remarque**

Un indicateur ou une commande à virgule flottante dont la gamme s’étend de –Infini à +Infini peut toujours générer une erreur de dépassement de gamme si vous lui envoyez NaN.

Correction des erreurs de dépassement de gamme des VIs

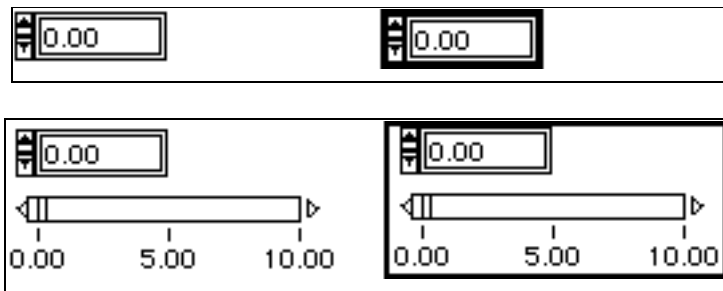


Indicateur d’erreur de dépassement de gamme

Un indicateur d’erreur de dépassement de gamme apparaît à la place du bouton **Exécution** dans les circonstances suivantes.

- Vous avez configuré une commande sur un sous-VI pour arrêter l’exécution lorsqu’elle reçoit une valeur hors gamme (par l’intermédiaire de l’élément de menu **Gamme des données...** de la commande), et la commande reçoit une telle valeur.
- Vous avez configuré un indicateur sur un sous-VI pour arrêter l’exécution lorsqu’il essaie de renvoyer une valeur hors gamme à un VI appelant, et l’indicateur a essayé de renvoyer une telle valeur.
- Un opérateur entre une valeur hors gamme dans une commande que vous avez configurée pour arrêter l’exécution en cas d’erreur, alors que le VI n’est pas en cours d’exécution.

Lorsqu'une commande ou un indicateur est hors gamme, une bordure épaisse rouge l'encadre, comme on peut le voir sur le côté droit de l'illustration suivante.



Pour plus d'informations, consultez la section *Options de gamme des commandes et indicateurs numériques* du chapitre 9, *Commandes et indicateurs numériques*.

 **Remarque**

Un VI ou sous-VI peut transmettre des valeurs hors gamme à l'un de ses indicateurs sans arrêter l'exécution. La condition d'erreur existe uniquement lorsqu'un sous-VI essaie de renvoyer une telle valeur. De même, un opérateur peut entrer des valeurs hors gamme dans une commande pendant l'exécution de son VI. La condition d'erreur de dépassement de gamme empêche seulement le VI de démarrer, mais pas de continuer une fois qu'il a démarré.

Fonctions de mise au point

Cette section aborde les fonctions de mise au point suivantes.

- Exécution d'un VI en mode pas à pas, afin d'observer chaque pas d'exécution
- Exécution en mode Animation, afin de visualiser le flux de données tel qu'il a lieu
- Utilisation d'une sonde pour afficher des données
- Mise en place de points d'arrêt pour suspendre l'exécution, afin que vous puissiez procéder à une exécution pas à pas, sonder des fils de liaison pour visualiser leurs données ou éditer des indicateurs
- Suspension de l'exécution pour éditer des indicateurs, contrôler le nombre de boucles d'exécution ou revenir au début du VI.

Exécution des VIs en mode Pas à pas



Bouton Pause

Pour les besoins de la mise au point, il peut être souhaitable d'exécuter un diagramme nœud par nœud. Cette forme d'exécution est appelée exécution en mode pas à pas. Avant d'exécuter un VI, commencez l'exécution pas à pas en appuyant sur le bouton **Pause**. L'exécution d'un VI peut aussi être suspendue à n'importe quel moment en cliquant sur le bouton **Pause**. Vous pouvez revenir à une exécution normale à n'importe quel moment en relâchant le bouton **Pause**. Le fait de cliquer sur le bouton **Exécution** lorsqu'un VI est en pause redémarre l'exécution, mais le VI reste en état de pause. Au prochain appel du VI, l'exécution est suspendue.

Bouton
Exécution
non détaillée

Pendant que le VI s'exécute en mode pas à pas, appuyez sur n'importe lequel des trois boutons Pas à pas actifs pour passer au pas suivant. Le bouton Pas à pas sur lequel vous appuyez détermine l'endroit où s'exécute le pas suivant.

Bouton
Exécution détaillée

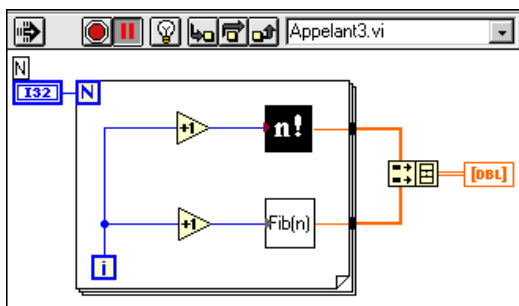
Si vous positionnez votre curseur sur l'un des boutons Pas à pas, une info-bulle apparaît avec une description du prochain pas qui sera exécuté si vous appuyez sur ce bouton.

Bouton
Sortie

Il peut être utile d'effectuer une exécution en mode Animation lorsque vous exécutez un VI en mode pas à pas, afin de suivre les données à mesure qu'elles circulent au travers des nœuds. Consultez la section *Exécution en mode Animation* de ce chapitre.

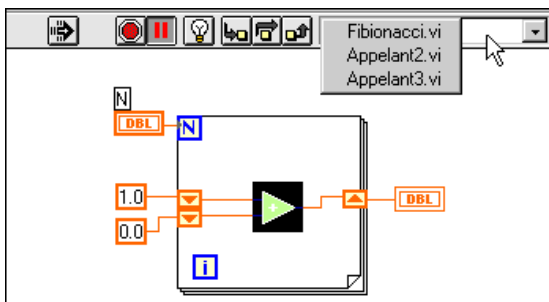
Exemple d'exécution d'un VI en mode Pas à pas

L'illustration suivante montre un exemple de VI en cours d'exécution pas à pas (VI appelant 1.vi). Ce VI s'exécute actuellement au nom du VI appelant 3.vi. Le sous-VI, Fibonacci.vi, est actuellement en cours d'exécution (indiqué par la flèche représentée sur le bouton **Exécuter**). Le VI, Factorial.vi, est le prochain nœud à s'exécuter. En cliquant sur le bouton **Exécution détaillée**, vous ouvrez son diagramme et vous commencez l'exécution en mode pas à pas. En cliquant sur le bouton **Exécution non détaillée**, vous exécutez le sous-VI et mettez le système en pause. En cliquant sur le bouton **Sortie**, vous finissez l'exécution du diagramme courant de la boucle et mettez le système en pause.



Après avoir exécuté de façon détaillée le sous-VI Fibonacci.vi et exécuté de façon détaillée la boucle, cliqué sur le bouton **Exécution non détaillée** et tenu le bouton de la souris enfoncé pendant une seconde, un menu apparaît. Sélectionnez jusqu'où vous voulez exécuter le VI avant de le mettre en pause. Le fait de sélectionner **Diagramme** exécute le VI jusqu'à l'exécution de tous les nœuds du diagramme, puis l'exécution est suspendue.

Si vous sélectionnez l'un des VIs dans la liste d'appel, ce VI s'exécute jusqu'à son terme. Ensuite, le VI appelant du VI sélectionné se met en pause.



Utilisation des boutons Pas à pas

Les boutons Pas à pas agissent comme suit.

Appuyez sur le bouton **Exécution non détaillée** pour exécuter une structure (séquence, boucle, etc.) ou un sous-VI et suspendre l'exécution au nœud suivant. Le raccourci clavier est <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) suivi de la flèche de déplacement vers la droite.



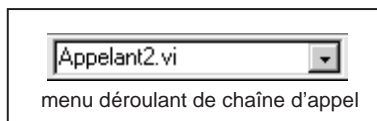
Bouton
Exécution détaillée

Appuyez sur le bouton **Exécution détaillée** pour exécuter le premier pas d'un sous-VI ou d'une structure (séquence, boucle, etc.) et suspendre l'exécution au pas suivant du sous-VI ou de la structure. Le raccourci clavier est <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) suivi de la flèche de déplacement vers le bas. Les boutons Pas à pas affectent l'exécution uniquement dans un VI ou un sous-VI en mode pas à pas. Si un VI en mode pas à pas contient un sous-VI également en mode pas à pas et un autre en mode d'exécution normale, le premier sous-VI s'exécute pas à pas lorsqu'il est appelé, tandis que le deuxième s'exécute normalement lorsqu'il est appelé. Appuyez sur le bouton **Sortie** pour finir d'exécuter le diagramme, la structure ou le VI en cours et suspendre l'exécution. Le raccourci clavier est <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) suivi de la flèche de déplacement vers le haut.

Lorsque l'exécution du VI se termine, les boutons Pas à pas deviennent grisés.

Lecture des chaînes d'appel

Lorsqu'un sous-VI est en pause, un menu déroulant **Chaîne d'appel** apparaît. Ce menu indique la chaîne des appelants, depuis le VI principal jusqu'à ce sous-VI. N'oubliez pas que ce menu ne contient pas les mêmes éléments que le menu **Projet»VIs appelant ce VI**, qui dresse la liste de tous les VIs appelants, qu'ils soient ou non en cours d'exécution. Lorsque vous sélectionnez un VI à partir du menu déroulant Chaîne d'appel, son diagramme s'ouvre et le VI appelant le sous-VI actuel est sélectionné. Ceci vous aide à distinguer l'instance actuelle du sous-VI si le diagramme contient plusieurs instances.



En plus des raccourcis clavier indiqués dans la section *Utilisation des boutons Pas à pas* de ce chapitre, les commandes suivantes sont disponibles :

- Lorsque vous exécutez un sous-VI ou une structure en mode pas à pas, le fait de cliquer sur le bouton **Exécution non détaillée** en maintenant enfoncé le bouton de la souris fait apparaître un menu. Sélectionnez jusqu'où vous voulez exécuter le VI avant de le mettre en pause.
- En double-cliquant sur les commandes et indicateurs de la face-avant, les variables locales et les variables globales, vous pouvez afficher et mettre en surbrillance l'objet du diagramme correspondant.
- Le fait d'appuyer sur la touche <Ctrl> (**Windows**), <option> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) pendant que vous double-cliquez sur un sous-VI déplace son diagramme au premier plan.

Exécution en mode Animation

Pour les besoins de la mise au point, il peut être utile de visualiser une animation de l'exécution du diagramme d'un VI.



arrêt

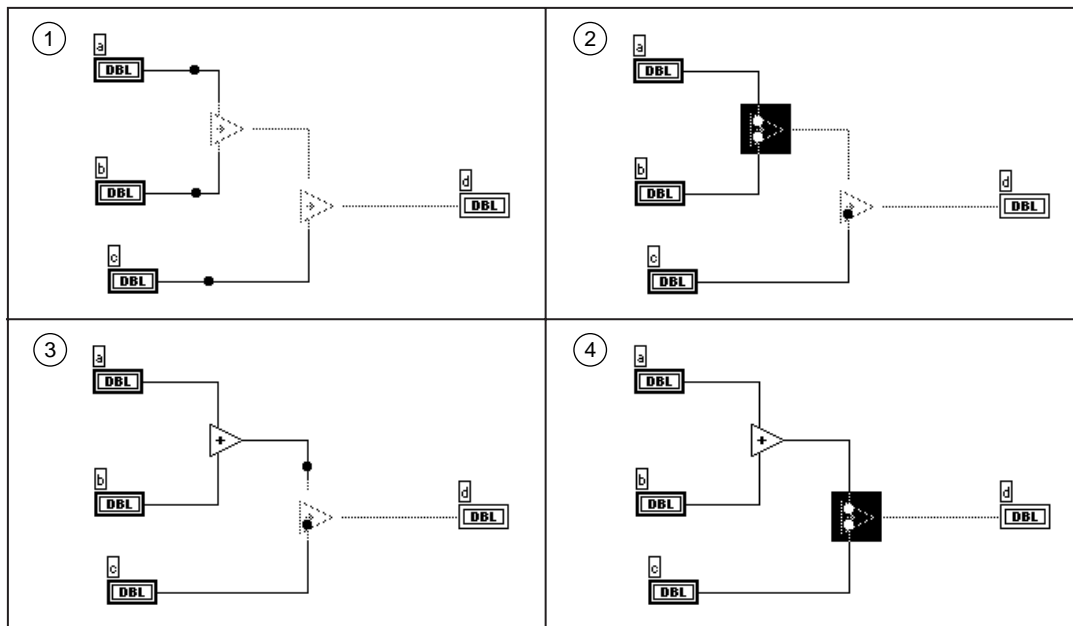


marche

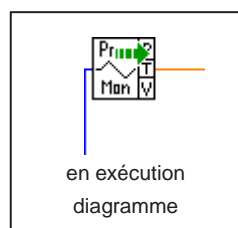
Bouton
Exécution en
mode Animation

Pour utiliser cette fonction, cliquez sur le bouton **Exécution en mode Animation**. Le bouton change d'apparence. Cliquez sur ce bouton à n'importe quel moment pour revenir en mode d'affichage normal. On utilise généralement l'exécution en mode Animation en conjonction avec le mode pas à pas pour mieux comprendre comment les données circulent au travers des nœuds. Le mode Animation réduit considérablement les performances d'un VI.

Avec l'exécution en mode Animation, le mouvement des données d'un nœud à un autre est marqué par des bulles se déplaçant le long des fils de liaison. De plus, en mode pas à pas, le nœud suivant clignote rapidement, comme le montre la séquence d'illustrations suivante.

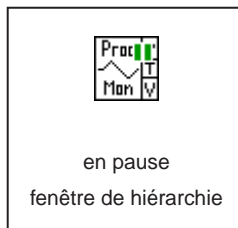


Lorsque vous exécutez un sous-VI en mode pas à pas avec le bouton **Exécution en mode Animation** sur marche, un glyphe d'exécution apparaît sur l'icône des sous-VIs du diagramme, indiquant les VIs qui sont en cours d'exécution et ceux qui attendent d'être exécutés. Sur l'illustration suivante, la flèche indique qu'un sous-VI en cours d'exécution attend d'être exécuté.



La fenêtre Hiérarchie indique si un VI est en pause et/ou interrompu (pour plus d'informations sur la fenêtre Hiérarchie, consultez la section *Utilisation de la fenêtre Hiérarchie* du chapitre 3, *Utilisation des sous-VIs*). Le glyphe de pause indique qu'un sous-VI est en pause et/ou interrompu. Un glyphe de pause vert (ou un glyphe vide, s'il est affiché en noir et blanc) indique que ce sous-VI se met en pause lorsqu'il est appelé. Un glyphe de

pause rouge (ou un glyphe plein, s'il est affiché en noir et blanc) indique que ce sous-VI est actuellement en pause.



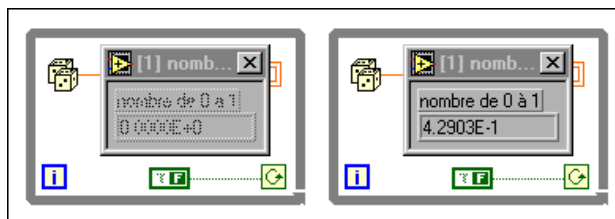
Utilisation de l'outil Sonde



Vous pouvez utiliser l'outil Sonde pour vérifier des valeurs intermédiaires dans un VI qui s'exécute mais produit des résultats douteux ou inattendus. Par exemple, vous pouvez avoir un diagramme compliqué avec une série d'opérations, une d'entre elles pouvant renvoyer des données incorrectes.

Une façon de chercher la source des résultats douteux consiste à câbler un indicateur au fil de liaison de sortie de l'une des opérations pour afficher les résultats intermédiaires. Mais placer un indicateur sur la face-avant et câbler ses terminaux au diagramme n'est pas un mécanisme pratique de mise au point. Cela prend du temps et ajoute sur votre face-avant et votre diagramme des éléments indésirables que vous devez ensuite supprimer.

Sélectionnez l'outil Sonde et placez son curseur sur un fil de liaison, ou ouvrez le menu local du fil de liaison et sélectionnez **Sonde**. Comme cela est représenté dans l'illustration suivante, la fenêtre flottante Sonde apparaît, et elle ne contient aucune donnée. Dès que vous exécutez le VI, la fenêtre Sonde affiche les données transmises le long du fil de liaison.



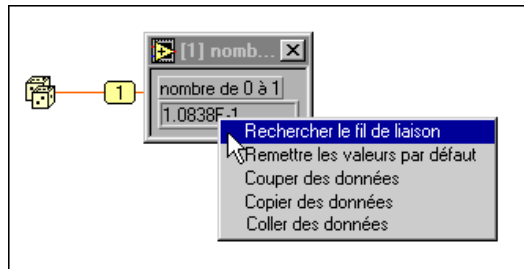
Chaque sonde est numérotée de façon automatique et unique. Le fil de liaison auquel elle est attachée est repéré avec le même numéro. Ceci permet d'identifier plus facilement la sonde qui est associée à un fil de liaison particulier. Cependant, si le nom de la commande de la face-avant est aussi long ou plus long que la fenêtre Sonde, le numéro n'est pas visible.

Vous pouvez utiliser la sonde en conjonction avec une exécution en mode Animation, une exécution pas à pas ou des points d'arrêt pour visualiser plus facilement certaines valeurs. Lors d'une exécution pas à pas ou d'une pause au niveau d'un point d'arrêt, les données sont mises à jour immédiatement si elles sont disponibles. Examinez les entrées lorsque l'exécution s'arrête à un nœud.

Vous pouvez insérer la sonde avant d'exécuter votre VI pour visualiser les données. En mode pas à pas, créez une sonde pour un fil de liaison exécuté. La sonde se met à jour, affichant le contenu du fil de liaison. Ceci est utile lorsque vous avez fixé un point d'arrêt sur un nœud et que vous voulez examiner ses entrées.

Vous ne pouvez pas changer les données avec la sonde, et la sonde n'a aucun effet sur l'exécution du VI.

Si vous ouvrez le menu local de l'indicateur de la fenêtre Sonde, vous pouvez trouver le fil de liaison associé en sélectionnant **Rechercher le fil de liaison**, comme le montre l'illustration suivante.



Si vous sélectionnez l'élément de menu **Rechercher le fil de liaison**, le diagramme contenant ce fil de liaison s'affiche au premier plan, devant tous les autres VIs, et le fil de liaison est mis en évidence.

Utilisez **Réinitialiser avec les paramètres par défaut** pour ramener la valeur affichée dans la fenêtre Sonde à sa valeur par défaut.

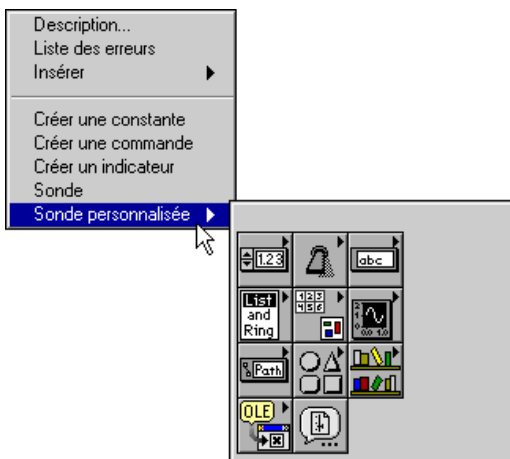
Utilisez l'élément de menu **Copier des données** pour copier les données vers d'autres commandes numériques dans le même VI ou dans d'autres VIs.

Création de sonde

Lorsque vous créez une sonde, son style s'adapte par défaut au type de données du fil de liaison. Si le fil de liaison transmet par exemple des données numériques, la sonde est alors un indicateur numérique.

Si vous préférez, sélectionnez une commande pour la sonde parmi des commandes intégrées ou parmi des commandes sauvegardées comme commandes personnalisées ou définitions de type. Pour ce faire, ouvrez le menu local du fil de liaison, sélectionnez **Sonde personnalisée**, puis sélectionnez une commande dans la palette **Sonde personnalisée** qui apparaît sur la droite. Si vous choisissez **Sélectionner une commande...**, utilisez la boîte de dialogue de fichier pour sélectionner toute commande

personnalisée ou définition de type sauvegardée dans le système de fichiers. Les définitions de types sont traitées comme des commandes personnalisées standard lorsqu'elles sont utilisées pour sonder des données.



Seules les parties de la palette **Sonde personnalisée** qui peuvent correspondre au type de données du fil de liaison sont activées. Si vous choisissez une commande qui ne peut pas avoir le même type de données que le fil de liaison, ce défaut de concordance est signalé par un bip et la sonde n'est pas créée. Par exemple, avec les tableaux et les clusters, vous ne pouvez pas utiliser les tableaux et clusters vierges de la palette **Tableaux et clusters**, parce que leur type de données n'est pas défini.

Ajoutez vos propres palettes en bas de la palette **Sonde personnalisée**, exactement comme vous les ajoutez à la palette **Commandes**. Pour plus

d'informations sur la méthode à employer, consultez le chapitre 7, *Personnalisation de votre environnement*.

Mise en place de points d'arrêt

Vous pouvez placer un point d'arrêt sur un VI, un nœud ou un fil de liaison pour qu'une pause ait lieu au cours de l'exécution. Les nœuds comprennent les sous-VIs, les fonctions, les structures, les Code Interface Nodes (CIN), les boîtes de calcul et les Attribute Nodes. Lorsqu'un point d'arrêt est affecté à un diagramme, l'exécution marque une pause une fois que l'exécution de tous les nœuds du diagramme est terminée. Lorsqu'un fil de liaison contient un point d'arrêt, l'exécution marque une pause une fois que des données ont été transmises par le fil de liaison.

Lorsque vous atteignez un point d'arrêt au cours de l'exécution, continuez l'exécution en mode pas à pas à l'aide des boutons Pas à pas, sondez les fils de liaison pour visualiser leurs données, changez les valeurs des commandes de la face-avant ou poursuivez l'exécution jusqu'au prochain point d'arrêt ou jusqu'à ce que l'exécution soit terminée. Cliquer sur le bouton **Exécution** est un moyen facile de continuer l'exécution jusqu'au point d'arrêt suivant ou jusqu'à ce que le sous-VI soit à nouveau appelé.

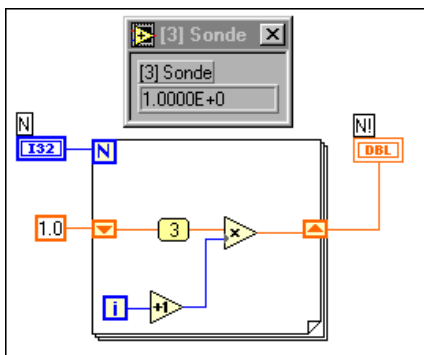
Pour placer un point d'arrêt, sélectionnez l'outil Point d'arrêt dans la palette **Outils**, puis cliquez sur le diagramme, le nœud ou le fil de liaison. Cliquez avec cet outil sur le même objet à n'importe quel moment pour supprimer le point d'arrêt. L'apparence de l'outil indique si un point d'arrêt est à placer ou à enlever, comme cela est représenté sur la gauche.

Le tableau suivant indique comment s'affichent les points d'arrêt et à quel moment s'opère la pause dans l'exécution, par rapport à l'endroit où sont placés les points d'arrêt.

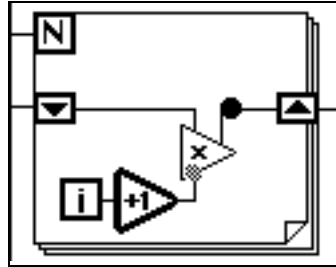
Tableau 4-2. Mise en place de points d'arrêt

Emplacement du point d'arrêt	Mise en évidence du point d'arrêt	Moment auquel la pause a lieu
Diagramme	Bordure rouge autour du diagramme. Si le diagramme est à l'intérieur d'une structure, la bordure rouge est également à l'intérieur de la structure.	Lorsque l'exécution de tous les nœuds du diagramme est terminée. Dans le cas des boucles, il y a une pause après chaque itération de la boucle.
Nœud	Bordure rouge encadrant le nœud.	Juste avant l'exécution du nœud. Ainsi, tous les signaux d'entrée dans le nœud peuvent être sondés avec l'outil Sonde.
Fil de liaison	Pastille rouge au milieu du fil de liaison. Si une sonde est branchée au fil de liaison, la sonde a une bordure rouge.	Après le passage des données dans le fil de liaison.

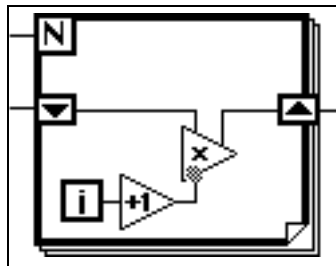
Dans l'exemple suivant, le fil de liaison, auquel est branchée une sonde, a un point d'arrêt. Le VI marque une pause une fois que la sonde affiche des données.



Dans l'exemple suivant, le nœud Incrément a un point d'arrêt. Le VI marque une pause avant l'exécution du nœud Incrément. Un fil de liaison a également un point d'arrêt. Le VI marque à nouveau une pause après l'exécution de Multiplier.



Dans l'exemple suivant, la boucle For a un point d'arrêt. Le VI marque une pause après l'exécution de Multiplier.



Lorsqu'un VI marque une pause à cause d'un point d'arrêt, son diagramme est amené au premier plan (LabVIEW ouvre automatiquement le face-avant VI si celle-ci était fermée), et le nœud ou le fil de liaison provoquant l'arrêt est mis en évidence avec une marqueuse.

Les points d'arrêt sont sauvegardés avec un VI, mais ils deviennent seulement actifs au cours de l'exécution.

Interruption de l'exécution

Vous pouvez éditer les commandes et les indicateurs, exécuter le sous-VI autant de fois que vous le désirez avant de revenir au VI appelant, ou retourner au début du sous-VI en interrompant l'exécution d'un sous-VI.

Pour mettre un sous-VI en mode d'interruption, ouvrez le sous-VI et cochez l'élément de menu **Exécution»Interrompre à l'appel**. Vous pouvez également accéder à cette fonction en ouvrant le menu local du

cadre connecteur sur la face-avant pendant que vous êtes en mode d'exécution, et en sélectionnant **Configuration du VI»Options d'exécution» Interrompre à l'appel...** Le sous-VI s'interrompt automatiquement lorsqu'il est appelé. Si vous cochez cet élément pendant une exécution pas à pas, le VI ne passe pas immédiatement en mode d'interruption.

Si vous voulez interrompre l'exécution au moment d'un appel particulier à un sous-VI, utilisez l'élément de menu **Configuration du nœud du sous-VI...** à partir du menu local du nœud du sous-VI, au lieu de **Interrompre à l'appel**. La **Configuration du nœud du sous-VI...** interrompt l'exécution à cette instance particulière du sous-VI uniquement.

Comment reconnaître une interruption automatique

Un VI s'interrompt automatiquement si, au cours de son exécution, une commande ou un indicateur sort de sa gamme. Le terminal ou la variable locale causant l'erreur de dépassement de gamme est mis en évidence.

Un sous-VI muni d'une commande ou d'un indicateur programmé pour s'arrêter en cas d'erreur de dépassement de gamme a un point d'arrêt conditionnel. Si une erreur de dépassement de gamme survient, le sous-VI marque une pause, comme s'il avait rencontré un point d'arrêt. Si aucune erreur de dépassement de gamme ne survient, l'exécution du sous-VI se déroule normalement.

Lorsque des données transmises à un sous-VI provoquent une erreur de dépassement de gamme, la face-avant du sous-VI s'ouvre ou est affichée au premier plan, et le sous-VI reste en mode d'interruption. A ce moment-là, les valeurs des commandes du sous-VI correspondent aux entrées transmises par le VI appelant, et vous pouvez changer les valeurs si vous le désirez. En fait, vous devez les changer pour exécuter le sous-VI si l'indicateur d'erreur de dépassement de gamme est allumé. Les indicateurs du sous-VI affichent soit les valeurs par défaut, soit les valeurs résultant de la dernière exécution du sous-VI au cours de laquelle la face-avant de celui-ci était ouverte.

Utilisation des boutons de la barre d'outils lorsque l'exécution des sous-VIs est interrompue



Bouton Exécuter

Si vous voulez exécuter le sous-VI en cours avant de revenir au VI appelant, appuyez sur le bouton **Exécuter** (ou sélectionnez la commande **Exécution»Exécuter**) pendant que vous êtes en mode d'interruption. Répétez l'exécution autant de fois que vous le voulez.

Lorsque l'exécution du sous-VI s'achève, les indicateurs affichent les résultats de cette exécution du sous-VI. Cependant, vous pouvez changer les valeurs des indicateurs si vous voulez renvoyer des valeurs différentes au VI appelant. En fait, vous ne pouvez fixer les valeurs des indicateurs du sous-VI que lorsque celui-ci est en mode d'interruption. Cliquez à nouveau sur le bouton **Revenir au VI appelant** lorsque vous êtes prêt à renvoyer les valeurs des indicateurs du sous-VI au VI appelant.



Retour au début

Si vous voulez revenir au début du VI, cliquez sur le bouton **Retour au début**.



Revenir au VI appelant

Le bouton **Revenir au VI appelant** apparaît lorsqu'un sous-VI interrompu ne s'exécute pas. Cliquez dessus pour revenir au VI appelant. Notez que vous pouvez revenir à un VI appelant sans exécuter le VI en cours. Si vous voulez l'exécuter, veillez à appuyer sur le bouton **Exécution** avant de revenir au VI appelant.

Visualisation de la fenêtre Hiérarchie pendant l'interruption



La fenêtre Hiérarchie affiche un glyphe en point d'exclamation pour indiquer un sous-VI interrompu. L'illustration suivante représente un sous-VI dans la fenêtre Hiérarchie, avec l'option **Interrompre à l'appel** activée.

Désactivation des fonctions de mise au point

Pour réduire la mémoire requise et pour augmenter légèrement ses performances, vous pouvez compiler un VI sans diverses fonctions de mise au point. Pour ce faire, allez dans **Configuration du VI»Options des fenêtres** à partir du menu local du cadre connecteur sur la face-avant et désélectionnez la case pour **Autoriser la mise au point**.

Désactivation de sections de diagrammes

Dans certains cas, il peut être souhaitable d'exécuter un VI avec une section du diagramme désactivée. La meilleure façon d'y arriver consiste à encadrer la section du diagramme que vous voulez désactiver dans une structure Condition avec une constante booléenne câblée au sélecteur. Mettez le diagramme que vous voulez désactiver dans le cadre de la structure Condition qui ne s'exécute pas.

Si la structure Condition doit produire des valeurs de données, créez des constantes pour les tunnels de sortie.

Impression et documentation des VIs

Ce chapitre aborde plusieurs questions concernant l'impression et la documentation des VIs.

Impression

Il y a quatre façons principales d'imprimer des VIs en G.

- Utilisez l'élément de menu **Imprimer la fenêtre...** pour effectuer une impression rapide du contenu de la fenêtre courante.
- Effectuez une impression plus complète d'un VI, comprenant des informations sur la face-avant, le diagramme, les sous-VIs, les commandes, l'historique du VI, et ainsi de suite, en sélectionnant l'élément de menu **Imprimer la documentation...**
- Utilisez la fonction d'impression contrôlée par programme pour que les VIs impriment automatiquement leur face-avant sous le contrôle de votre application.
- Utilisez le VI Serveur pour imprimer une fenêtre de VI ou la documentation d'un VI à partir d'un programme.

De plus, vous pouvez utiliser les VIs Port série pour envoyer du texte à l'imprimante si celle-ci est branchée à un port série ou parallèle ; pour employer cette méthode, il est généralement nécessaire de posséder quelques connaissances du langage de commande de l'imprimante.

Sous Windows et UNIX, une autre option pour imprimer du texte consiste à utiliser le VI "Exec. système" (System Exec.vi) qui se trouve dans la palette **Fonctions»Communication**. Sur Macintosh, utilisez le VI "AESend Print Document" qui se trouve dans la palette **Fonctions»Communication»AppleEvent**. Sous Windows, utilisez ActiveX Automation pour indiquer à une autre application d'imprimer des données.

Configuration de l'impression

Deux boîtes de dialogue affectent l'apparence de toutes les impressions, quelle que soit la méthode que vous utilisez pour imprimer : la boîte de dialogue Préférences et la boîte de dialogue Configuration.

Vous utilisez la page de dialogue Impression de la boîte de dialogue Préférences pour configurer les options d'impression.

Pour configurer le formatage et les informations spécifiques à une imprimante, utilisez l'élément de menu **Configuration de l'imprimante...** (**Mise en page** sur Macintosh) dans le menu **Fichier**. Par exemple, avec la plupart des imprimantes, vous pouvez changer l'orientation de l'impression (paysage ou portrait) à l'aide de cette boîte de dialogue. De nombreuses imprimantes ont également des options pour la substitution des polices, la taille du papier et d'autres réglages spécifiques à l'imprimante. Les choix que vous faites dans **Configuration de l'imprimante...** (**Mise en page** sur Macintosh) sont sauvegardés avec votre VI.

Impression en PostScript

Si vous avez accès à une imprimante PostScript, vous pouvez choisir l'élément **Edition»Préférences»Impression»Impression PostScript** plutôt que d'utiliser d'autres options d'impression.

Une imprimante PostScript offre les avantages suivants :

- Les impressions en PostScript reproduisent plus fidèlement l'image de l'écran.
- PostScript facilite les graphiques à haute résolution.
- PostScript reproduit plus fidèlement les motifs, les styles de ligne et les polices de caractères.

Vous pouvez utiliser la boîte de dialogue Préférences pour indiquer si vous voulez utiliser l'impression en PostScript. Pour plus d'informations sur la boîte de dialogue Préférences, consultez le chapitre 7, *Personnalisation de votre environnement*.

Impression de la fenêtre active

Utilisez l'élément de menu **Imprimer la fenêtre...** pour imprimer le contenu de la fenêtre active (face-avant ou diagramme). A l'aide de cet élément de menu, vous pouvez effectuer une impression rapide avec un

nombre minimum de confirmations. Pour des impressions plus complètes des VIs, utilisez **Imprimer la documentation...**

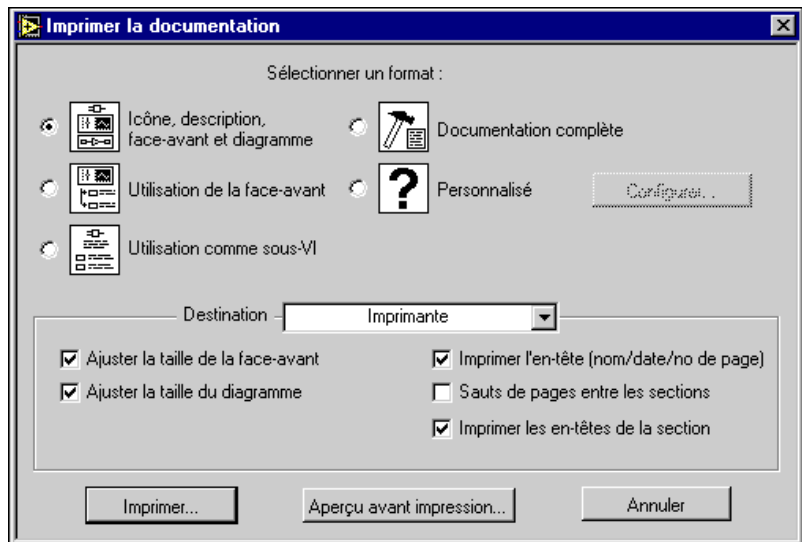
Imprimer la fenêtre... formate l'impression pour qu'elle ressemble exactement à ce que vous obtenez lorsque vous configurez le VI avec l'option d'impression contrôlée par programme. La section **Options d'exécution** de la boîte de dialogue de configuration des VIs contient quelques options qui vous donnent plus de contrôle sur l'apparence de votre VI lorsque vous effectuez une impression contrôlée par programme ou en utilisant **Imprimer la fenêtre...** Pour plus d'informations sur ces options, consultez la section *Définition de la mise en page*, plus loin dans ce chapitre.

Documentation des VIs

Cette section décrit comment imprimer la documentation de vos VIs, sur papier ou dans des fichiers RTF ou HTML, et donne des informations relatives au formatage des fichiers graphiques.

Impression de la documentation

Si vous voulez une impression détaillée du contenu d'un VI, utilisez **Imprimer la documentation...**, comme le montre l'illustration suivante. Utilisez la boîte de dialogue pour choisir parmi plusieurs formats de VI. Vous pouvez également personnaliser votre propre format.



Définition des formats d'impression

La boîte de dialogue **Imprimer la documentation** contient cinq formats d'impression différents, ainsi qu'une option personnalisée pour créer votre propre format. L'icône correspondant à chacun de ces formats est représentée à gauche de sa description.



Le format **Icône, description, face-avant et diagramme** (le format par défaut) imprime l'icône, la description du VI, la face-avant et le diagramme.



Le format **Utilisation de la face-avant** imprime la face-avant, la description du VI, le nom des commandes et leur description.



Le format **Utilisation comme sous-VI** imprime l'icône, le connecteur, la description du VI, les terminaux (types de données), le nom et la description des commandes connectées. Ce format est similaire au format de référence de fonction G.



Le format **Documentation complète** imprime tout, y compris l'icône, le connecteur, la description, la face-avant, des informations concernant toutes les commandes de la face-avant, le diagramme et une liste des noms des sous-VIs.



Le format **Personnalisé** imprime en utilisant les paramètres personnalisés actuellement en vigueur. Changez les paramètres personnalisés en sélectionnant **Configurer...** une fois que vous avez sélectionné l'élément de menu de personnalisation. Pour plus d'informations sur cette boîte de dialogue, consultez la section *Création de paramètres d'impression personnalisée* dans ce chapitre.

La boîte de sélection **Destination** vous permet de choisir entre l'imprimante, un fichier HTML, un fichier RTF ou un fichier texte ordinaire. Pour plus d'informations sur les fonctions **Destination**, consultez la section *Imprimer/Exporter des descriptions de commandes et de VIs vers un fichier RTF ou HTML* de ce chapitre.

Définition d'autres options d'impression

La boîte de dialogue **Imprimer la documentation** contient plusieurs options de mise en page permettant de contrôler l'échelle, les sauts de page et les en-têtes pour l'impression. Celles-ci sont décrites ci-après.

Ajuster la taille de la face-avant et **Ajuster la taille du diagramme** : lorsqu'elle est définie par l'intermédiaire de cet élément de menu, l'échelle de la face-avant et/ou du diagramme peut être réduite jusqu'à un quart de la taille originale pour mieux tenir sur les pages.

Imprimer un en-tête : imprime un en-tête en haut de chaque page. Cet en-tête comprend le numéro de page, le nom du VI et la date de la dernière modification du VI.

Sauts de page entre sections : insère un saut de page entre les sections suivantes :

- Icône, connecteur et description
- Face-avant
- Liste de détails des commandes de la face-avant
- Diagramme
- Détails du diagramme
- Hiérarchie du VI
- Liste des sous-VIs

Imprimer les en-têtes des sections : imprime un en-tête pour chaque section (par exemple, un en-tête tel que Liste des sous-VIs avant les informations relatives aux sous-VIs). Lors d'une impression à l'aide de la fonction **Imprimer la documentation**, cette sélection est automatiquement activée pour chaque format.

Création de paramètres d'impression personnalisée

Pour faire apparaître la boîte de dialogue **Paramètres d'impression personnalisée**, sélectionnez le bouton **Personnalisé** dans la boîte de dialogue **Imprimer la documentation**. Une fois que vous avez sélectionné **Personnalisé**, le bouton **Configurer** est activé. Cliquez sur ce bouton pour ouvrir la boîte de dialogue représentée dans l'illustration suivante.



Cette boîte de dialogue affiche les catégories d'éléments que vous pouvez imprimer. Ces catégories sont classées dans l'ordre selon lequel elles apparaissent sur la copie imprimée. En plus, les options de mise en page de cette boîte de dialogue contrôlent l'échelle, les sauts de page et les en-têtes pour l'impression.

Une icône pour chacun des éléments de paramètre d'impression personnalisée est représentée à gauche de sa description.



Icône et description

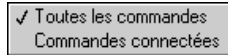
Connecteur et icône du VI : imprime une image de l'icône du VI avec ses entrées et sorties.

Description du VI : imprime la description du VI.



Face-avant : imprime la face-avant.

Commandes : imprime une liste des noms des commandes et indicateurs. Lorsque le système rencontre un tableau, un cluster ou un refnum, il imprime les sous-commandes. Sélectionnez les éléments de menu suivants à partir du menu local.



Descriptions : imprime les descriptions en face du nom des commandes.

Inclure des informations sur le type de données : si vous imprimez vers l'imprimante, le terminal de chaque commande est imprimé à gauche du nom de la commande. Si vous effectuez une sauvegarde dans un fichier texte, le type de données est imprimé après la description.



Diagramme : imprime le diagramme.

Cadres masqués : imprime les cadres non visibles des structures Condition et des structures Séquence.

Répéter les étapes de niveau supérieur : lors de l'impression de cadres non visibles, imprime à nouveau ceux qui sont visibles, dans l'ordre.



Hiérarchie du VI : imprime une description de la hiérarchie du VI actuellement en mémoire, avec des lignes représentant les connexions entre les VIs et leurs sous-VIs. Le VI courant est mis en évidence au sein d'une boîte.



Liste des sous-VIs : imprime l'icône, le nom et le chemin de tous les sous-VI utilisés.



Historique du VI : imprime les informations concernant l'historique du VI courant, s'il y en a.

Impression contrôlée par programme

Si vous voulez imprimer des éléments sous le contrôle de votre VI, plutôt que de façon interactive comme c'est le cas avec les boîtes de dialogue

Imprimer la fenêtre et Imprimer la documentation, il y a deux manières de procéder.

- A l'aide de l'impression contrôlée par programme, vous pouvez configurer un VI pour qu'il imprime automatiquement sa face-avant à chaque fois qu'il termine son exécution.
- A l'aide du VI Serveur, vous pouvez imprimer la face-avant ou la documentation de n'importe quel VI.

Pour utiliser l'impression contrôlée par programme, sélectionnez **Exécution»Imprimer à la fin de l'exécution**.

Une fois cette option activée, le VI imprime le contenu de la face-avant à chaque fois qu'il finit de s'exécuter. Si le VI est un sous-VI, il imprime la face-avant à la fin de son exécution, avant de revenir au VI appelant.

Contrôle du moment auquel ont lieu les impressions

Dans certains cas, vous ne souhaitez pas qu'un VI effectue une impression à chaque fois. Vous pouvez vouloir que cela se produise uniquement si l'utilisateur appuie sur un bouton, ou si une condition quelconque est remplie, comme par exemple si un test échoue. Vous pouvez également désirer davantage de contrôle sur le format d'impression. Ou vous pouvez vouloir imprimer seulement un sous-ensemble des commandes, ou même juste une commande spécifique.

Vous pouvez créer un sous-VI dont la face-avant est formatée pour avoir l'apparence que vous voulez pour le VI. Au lieu de sélectionner **Exécuter»Imprimer à la fin de l'exécution** sur votre VI, effectuez cette sélection à partir du sous-VI. Lorsque vous voulez une impression, vous pouvez appeler le sous-VI et lui transmettre les données à imprimer.

Avec cette méthode, vous avez un contrôle total sur l'apparence de l'impression, grâce à des outils et à des commandes G que vous pouvez utiliser pour créer des impressions très simples ou complexes.

Une autre méthode pour contrôler les circonstances dans lesquelles a lieu l'impression consiste à utiliser les méthodes d'impression du VI Serveur. Avec ces options, vous pouvez imprimer n'importe quel VI à n'importe quel moment. Pour plus d'informations, consultez le chapitre 21, *VI Serveur*.

Affinage des impressions

En utilisant la transparence et l'outil Pinceau, vous pouvez cacher les portions des commandes que vous ne souhaitez pas voir apparaître sur vos impressions. Par exemple, utilisez la transparence pour simplifier l'apparence des bords des commandes.

Utilisez les objets graphiques de la palette **Décorations** pour mettre en évidence des sections de votre impression. Par exemple, vous pouvez entourer une section de commandes à l'aide d'un cadre pour la faire ressortir du reste de votre face-avant.

Vous pouvez également utiliser des graphiques bitmap pour personnaliser votre impression en ajoutant à votre rapport des éléments tels que le logo de votre société.

Définition de la mise en page

Les paramètres qui sont fixés dans **Edition»Préférences** et dans **Configuration de l'imprimante (Mise en page sur Macintosh et UNIX)** affectent l'apparence de vos impressions. Utilisez **Configuration du VI»Options d'exécution** pour mettre en œuvre ou désactiver certaines options de mise en page affectant l'apparence de votre impression. Ces options de mise en page, représentées dans l'illustration suivante, affectent également le comportement de la fonction **Imprimer la fenêtre...**

The screenshot shows a dialog box with the following options and settings:

- Imprimer la face-avant à la fin de l'exécution du VI
- Imprimer l'en-tête (nom, date, no de page)
- Ajuster pour l'impression
- Entourer la face-avant d'une bordure lors de l'impression
- Marges des pages
 - Unité: pouce, mm
 - Haut: 0.00
 - Gauche: 0.00, Droite: 0.00
 - Bas: 0.00

L'élément **Imprimer la face-avant à la fin de l'exécution du VI** est une autre manière d'activer l'impression contrôlée par programme.

Si vous sélectionnez **Imprimer un en-tête**, un en-tête contenant le nom du VI, la date de la dernière modification et le numéro de la page apparaît en haut de chaque page.

Si vous sélectionnez **Ajuster l'échelle**, et si la face-avant est plus grande qu'une simple page, l'échelle de l'impression est réduite jusqu'à un quart

de la taille originale pour faire tenir la face-avant sur un nombre minimum de pages.

Si vous sélectionnez **Encadrer la face-avant d'une bordure**, un cadre s'imprime autour de la face-avant.

Si vous sélectionnez **Marges de page**, vous pouvez fixer des marges absolues pour les impressions, en pouces ou en millimètres. Les quatre marges peuvent être définies séparément. Les marges sont limitées par les paramètres physiques de l'imprimante. Si vous essayez de fixer des marges plus étroites que ce qui est permis par l'imprimante, les marges effectivement utilisées sont les marges minimales permises par l'imprimante.

Autres méthodes d'impression

Il se peut que la méthode d'impression mentionnée ci-dessus ne convienne pas à votre application. Il y a certaines techniques supplémentaires qui répondent à des besoins d'impression particuliers.

Par exemple, la méthode d'impression exposée plus haut vise à imprimer une page de données entière. Dans certaines applications, il peut être préférable d'imprimer des données ligne par ligne. Si une imprimante à impression en ligne est branchée à votre port série ou parallèle, utilisez les VIs Port série pour envoyer du texte à l'imprimante. Pour ce faire, il faut généralement quelques connaissances du langage de commande de l'imprimante, mais cela a donné d'excellents résultats pour de nombreuses applications développées par des utilisateurs du G.

Si les résultats imprimés ne sont pas ce que vous souhaitez, envisagez d'utiliser une autre application pour imprimer vos données, en sauvegardant les données dans un fichier et en imprimant ensuite à partir de l'autre application.

Pour plus de détails sur les méthodes d'impression alternatives, consultez la section *Comment puis-je imprimer une chaîne de caractères ?* de l'Annexe B, *Questions courantes concernant le G*.

Imprimer/Exporter des descriptions de commandes et de VIs vers un fichier RTF ou HTML

Vous pouvez imprimer ou exporter des descriptions de VIs et de commandes dans des fichiers de format RTF (*Rich Text Format*) ou HTML (*Hyper Text Markup Language*). Ceci est utile dans la mesure où vous pouvez importer un fichier RTF dans la plupart des logiciels de publication

de documents. Les fichiers RTF constituent également la source pour les fichiers d'aide en ligne. Utilisez le format HTML pour les documents en ligne, en particulier ceux que vous avez l'intention de publier sur le World Wide Web.

Lorsque vous imprimez de la documentation dans un fichier RTF, vous pouvez préciser si vous voulez créer un fichier susceptible d'être utilisé comme fichier d'aide en ligne (référence en ligne) ou pour du traitement de texte. Dans le format destiné au traitement de texte, les graphiques sont placés dans le document. Lorsque vous imprimez votre documentation en format RTF pour des fichiers d'aide en ligne, les graphiques sont enregistrés dans des fichiers bitmap (.BMP) externes. Pour la documentation HTML, tous les graphiques sont enregistrés dans des fichiers externes en format JPEG (.JPG) ou Portable Network Graphics (.PNG).

Pour imprimer votre documentation au format RTF ou HTML, sélectionnez **Fichier»Imprimer la documentation**. La boîte de dialogue suivante apparaît.

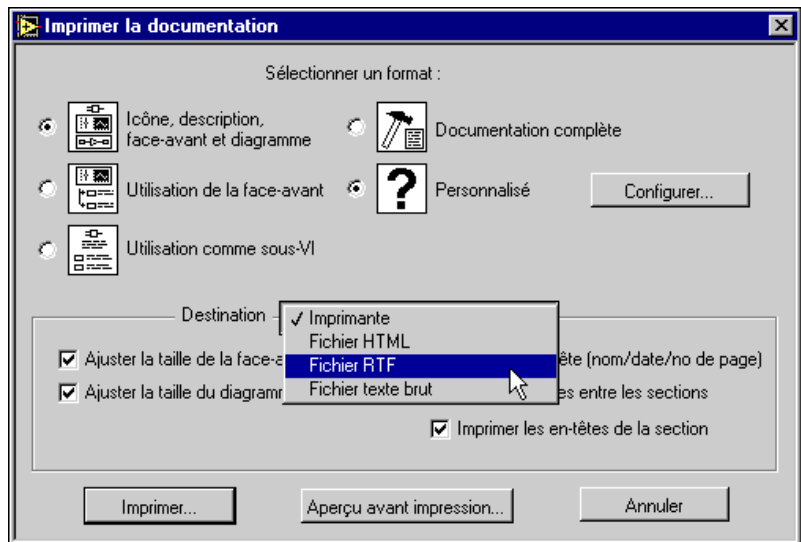


Figure 5-1. Boîte de dialogue Imprimer la documentation

Sélectionnez en tant que **Destination** un Fichier RTF ou un Fichier HTML.

Si vous sélectionnez Fichier RTF, la boîte de dialogue affiche les éléments de menu supplémentaires suivants. Enregistrez votre fichier au format d'aide en ligne en sélectionnant l'option **Source du compilateur de l'aide (images écrites en externe)**, qui enregistre les graphiques dans un fichier externe. Sélectionnez la **Destination** et le nombre de **Couleurs**.

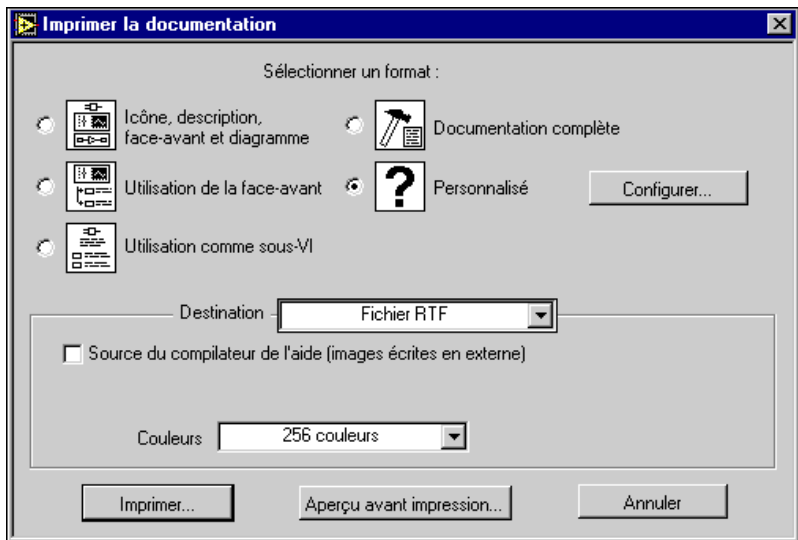


Figure 5-2. Boîte de dialogue Imprimer la documentation, Fichier RTF

Si vous sélectionnez Fichier HTML comme format pour votre documentation de commandes et de VIs, la boîte de dialogue apparaît comme ci-dessous.

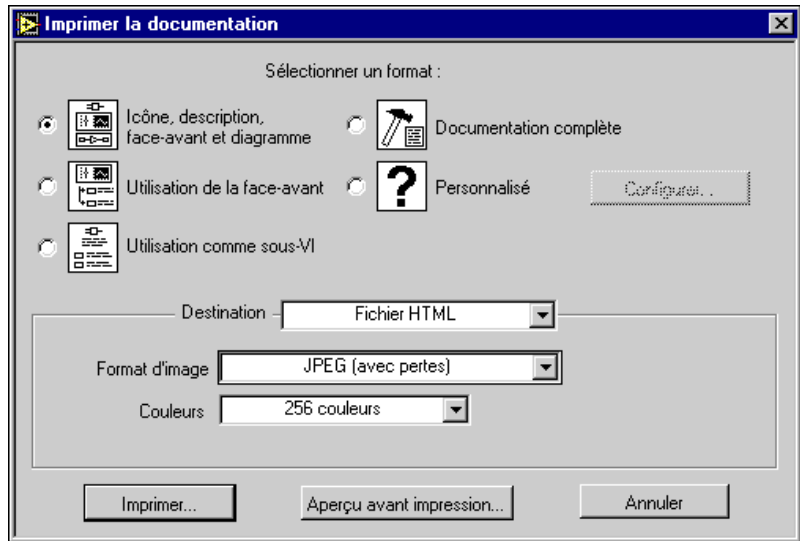


Figure 5-3. Boîte de dialogue Imprimer la documentation, Fichier HTML

Vous pouvez sélectionner le format des fichiers graphiques, ainsi que le nombre de couleurs. Le format JPEG offre une bonne compression, mais il peut perdre certains détails de l'image. Ce format convient mieux pour des photos. Pour des dessins, des schémas des faces-avant et des diagrammes, la compression JPEG peut donner lieu à une perte de netteté des images et d'uniformité des couleurs. Les images en JPEG sont toujours en 24 bits. Si vous choisissez un nombre de couleurs réduit avec JPEG, par exemple noir et blanc, la capture des images s'effectue avec le nombre de couleurs requis, par exemple noir et blanc, mais le résultat reste une image en 24 bits.

Le format PNG offre également une bonne compression, bien qu'elle ne soit pas toujours aussi bonne que celle du format JPEG. Cependant, la compression PNG ne perd aucun détail. De plus, ce format tolère des images en 1 bit, 4 bits, 8 bits et 24 bits. Pour un nombre de couleurs réduit, la compression de l'image résultante est bien meilleure que celle obtenue avec JPEG. Le format PNG a été développé pour remplacer le format GIF. Bien qu'il présente de nombreux avantages par rapport aux formats JPEG et GIF, son usage n'est pas aussi répandu.

Si vous voulez que les graphiques de vos fichiers HTML soient enregistrés au format GIF, vous pouvez convertir les graphiques en utilisant l'un des partageurs qui convertissent les formats graphiques. Dans ce cas, modifiez HTML pour faire référence aux images GIF avec l'extension correcte. Si vous choisissez de produire des images GIF de cette manière, nous vous conseillons de commencer par des images PNG parce qu'elles sont des reproductions sans pertes des graphiques originaux. Ensuite, convertissez ceux-ci en GIF. Pour des questions de licences, GIF n'est pas directement supporté, mais pourra l'être ultérieurement.

Lorsque vous enregistrez de la documentation pour le VI, une boîte de dialogue apparaît, vous demandant le nom du document. Le nom par défaut de toute documentation que vous enregistrez au format RTF ou HTML comprend le nom du VI avec, respectivement, une extension `.rtf` ou `.html` ajoutée au nom du fichier (comme `AI Clear.rtf`) (toute documentation que vous enregistrez en HTML sur la plate-forme Windows 3.1 comprend le nom du VI avec une extension `.htm`.)

Basé sur un exemple, le tableau suivant dresse la liste des paramètres du VI et des noms de fichiers JPEG qui leur sont associés. Si vous sélectionnez un format graphique JPEG lorsque vous enregistrez un VI nommé Test, vous obtenez les fichiers JPEG suivants.

Tableau 5-1. Fichiers JPEG résultants

Paramètre du VI	Nom du fichier JPEG
Icône du VI	<code>testi.jpg</code>
Connecteur du VI	<code>testc.jpg</code>
Face-avant du VI	<code>testp.jpg</code>
Diagramme du VI	<code>testd.jpg</code>
Fenêtre de hiérarchie du VI	<code>testh.jpg</code>
Icônes des sous-VIs	<code>nom du sous-VI + i.jpg</code>

Les images des terminaux de commandes et d'indicateurs sont enregistrées dans des fichiers d'images nommés en conséquence. Ainsi, si votre VI a plusieurs entrées du même type, vous obtenez un fichier d'images unique pour chacun des terminaux. Par exemple, si votre VI a trois entrées `int32`, un seul fichier `ci32.jpg` est créé.

L'attribution des noms de fichiers s'effectue de la façon suivante. La première lettre est un `c` ou un `i`, selon que le terminal correspond à une

commande ou à un indicateur. Si le terminal représente un tableau, la lettre est suivie par le nombre de dimensions du terminal (par exemple, 2d). Ensuite vient un court nom représentant le type de données. Ci-après se trouvent quelques exemples de noms de fichiers d'images.

Tableau 5-2. Exemples de noms de fichiers d'images

Nom	Terminal
ci32.jpg	commande, int32
i2di32.jpg	indicateur, tableau à 2 dimensions de int 32
c3dstr.jpg	commande, tableau à 3 dimensions de chaînes de caractères

Fichiers RTF et HTML contrôlés par programme

Pour plus d'informations sur la création de fichiers RTF et HTML gérée par programme, consultez le chapitre 21, *VI Serveur*. Ce chapitre contient également des informations sur l'exportation et l'importation de chaînes de caractères pour plusieurs VIs.

Localisation

En localisant les chaînes de caractères sur la face-avant d'un VI, vous pouvez aussi localiser les impressions du VI. Pour plus d'informations sur la localisation des chaînes de caractères des VIs, consultez la section *Localisation des VIs* du chapitre 29, *Portabilité et localisation*.

Création de vos propres fichiers d'aide

Avec les bons outils de développement, vous pouvez créer vos propres documents d'aide ou de référence en ligne. Les documents d'aide sont basés sur des documents de texte formatés, et vous pouvez entrer des sujets dans ces documents pour créer des connexions avec vos VIs.

Les fichiers source pour toutes les plates-formes doivent être au format Aide de Windows. Il est possible de créer des documents d'aide pour de multiples plates-formes.

Lorsque vous créez des documents source, vous utilisez un compilateur d'aide pour créer un document d'aide. Si vous voulez des fichiers d'aide sur plusieurs plates-formes, vous devez utiliser le compilateur d'aide pour la

plate-forme spécifique sur laquelle vous avez l'intention d'utiliser le fichier. Vous pouvez utiliser n'importe lequel des compilateurs suivants. Les compilateurs Windows comprennent également des outils pour créer des documents d'aide.

- **(Windows)** RoboHelp de Blue Sky Software, 800 677 4946 ; pour les clients internationaux, 619 459 6365.
- **(Windows)** Doc-To-Help de WexTech Systems, Inc., 800 939 8324.
- **(Macintosh)** QuickView de Altura Software, 408 655 8005.
- **(UNIX)** HyperHelp de Bristol Technologies, 203 438 6969.

Une fois que vous avez créé et compilé vos fichiers d'aide, vous pouvez les relier directement à un VI. Ouvrez le menu local sur le cadre connecteur du VI auquel vous voulez relier un fichier d'aide et sélectionnez

Configuration du VI»Documentation. Sélectionnez la boîte **Repère de l'aide** et tapez le lien du sujet dans le fichier d'aide. Choisissez le fichier d'aide en cliquant sur le bouton **Parcourir...** Le chemin du fichier apparaît dans la boîte **Chemin de l'aide**.

Configuration des VIs et sous-VIs

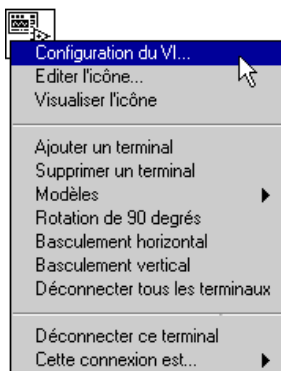
Ce chapitre traite de la personnalisation du comportement des VIs grâce aux boîtes de dialogue **Configuration du VI** et **Configuration du nœud du sous-VI**.

Création de panneaux locaux

Une face-avant unique est parfois trop restrictive pour présenter de nombreux affichages ou options. Pour y remédier, organisez vos VIs de manière à ce que le VI principal présente les options de haut niveau et les sous-VIs des options connexes.

Lorsque le G fait appel à un sous-VI, il l'exécute généralement sans ouvrir sa face-avant. Vous pouvez utiliser les boîtes de dialogue **Configuration du VI** ou **Configuration du nœud du sous-VI** pour faire en sorte que le sous-VI ouvre sa face-avant lors de l'appel et la ferme en fin d'exécution.

Vous pouvez accéder à la boîte de dialogue **Configuration du VI** en ouvrant un menu local de l'icône du VI en haut à droite de sa face-avant et en sélectionnant l'option **Configuration du VI...**, comme indiqué dans l'illustration suivante. Vous devez être en mode Edition.



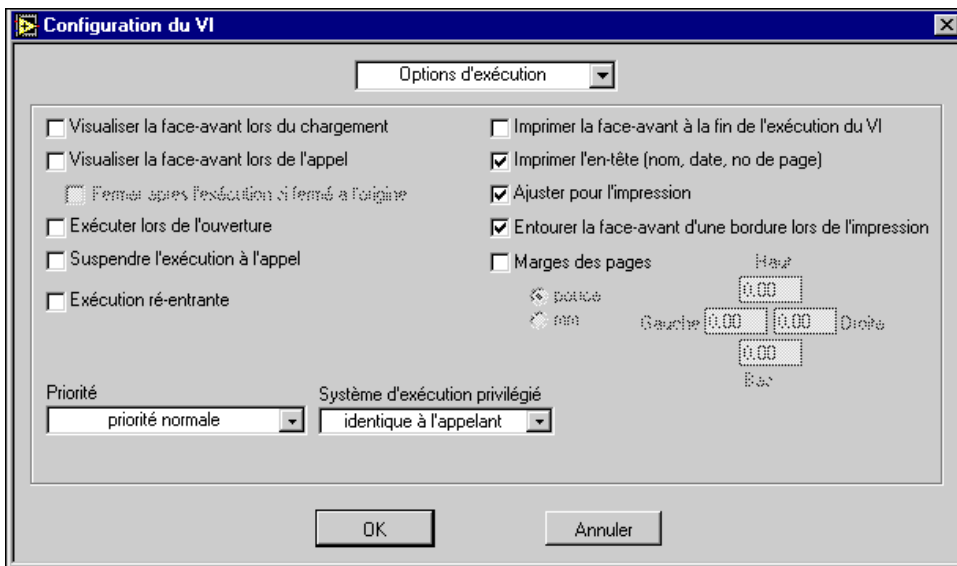
Cette boîte de dialogue contient plusieurs options de personnalisation de l'aspect et du comportement du VI. Utilisez le menu déroulant du haut de la boîte de dialogue pour sélectionner une des quatre catégories d'options : **Options d'exécution**, **Options des fenêtres** et **Documentation**.

Options de configuration des VIs

Cette section décrit les options d'exécution, du contexte visuel, de la documentation, et de personnalisation du menu de vos VIs.

Options d'exécution

Au départ, la boîte de dialogue **Configuration du VI** vous présente les options d'exécution du VI en cours. Les **Options d'exécution** sont répertoriées dans l'illustration suivante.



Si vous souhaitez que la face-avant du sous-VI s'ouvre dès le chargement du VI principal, activez l'option **Visualiser la face-avant lors du chargement**.

Les deux options les plus utiles de cette page sont **Visualiser la face-avant lors de l'appel** et **Fermer après l'exécution si fermé à l'origine**. Si vous activez ces options pour un sous-VI, sa face-avant s'ouvrira automatiquement lors de l'appel du sous-VI et se fermera automatiquement, s'il était fermé à l'origine, à la fin de son exécution.

L'option **Exécuter lors de l'ouverture** lance automatiquement l'exécution d'un VI lors de son ouverture.

Le fait d'activer l'option **Interrompre à l'appel** revient à effectuer **Exécuter»Interrompre à l'appel**. Cette option de mise au point est décrite dans la section *Fonctions de mise au point* du chapitre 4, *Exécution et mise au point des VIs et sous-VIs*.

Exécution ré-entrante, **Priorité** et **Système d'exécution préféré** sont des fonctions plus avancées qui affectent l'exécution du VI. Elles ne vous serviront que dans des applications spéciales. Ces options sont respectivement décrites dans les sections *Vue d'ensemble de l'exécution ré-entrante* et *Priorité dans la configuration du VI* du chapitre 26, *Comprendre le système d'exécution du G*. Une autre fonction d'exécution avancée est le *multithreading*. Dans cette dernière, l'exécution simultanée de plusieurs VIs a lieu alors que des VIs répondent toujours aux entrées de la souris et du clavier et l'horloge du CPU est partagée de manière égale entre les threads d'exécution. Cette fonction est également décrite dans le chapitre 26, *Comprendre le système d'exécution du G*.

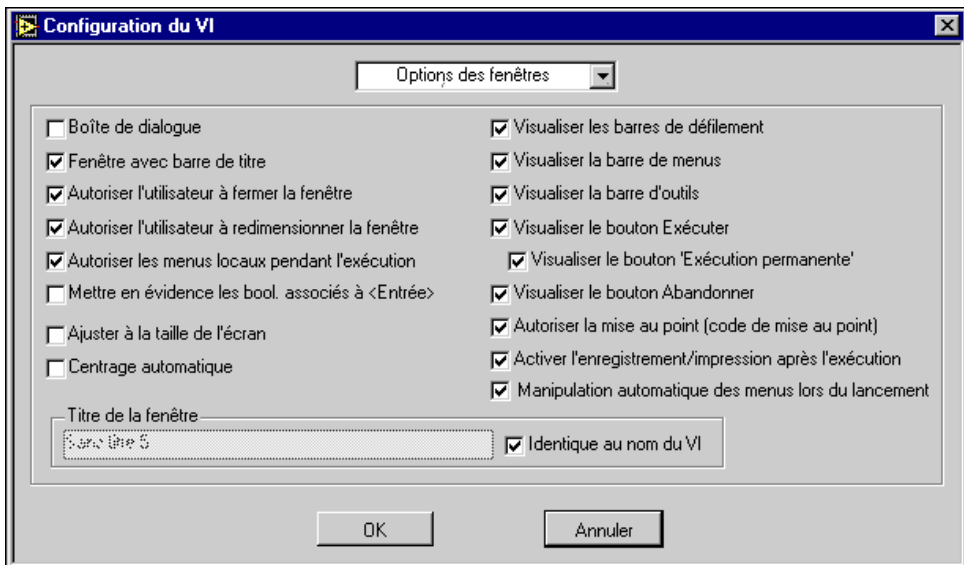
Utilisez les autres options pour imprimer votre VI après exécution et personnaliser l'impression. L'impression contrôlée par programme est traitée dans le détail dans la section *Impression contrôlée par programme* du chapitre 5, *Impression et documentation des VIs*.

Options des fenêtres

Les options des fenêtres s'appliquent au VI lors de son exécution, mais pas en mode Edition. Utilisez ces options afin de contrôler la capacité de l'utilisateur à agir sur le programme en restreignant l'accès aux fonctions du G et en changeant l'aspect et le comportement de la fenêtre. Votre VI peut par exemple se comporter comme une boîte de dialogue en ne permettant pas à l'utilisateur d'agir sur d'autres fenêtres lorsque le VI est ouvert. Vous pouvez également supprimer les barres de défilement et la barre d'outils, ainsi que centrer une fenêtre ou l'ajuster de manière automatique à l'écran.

Vous pouvez personnaliser le titre de la fenêtre du VI pour qu'il soit plus descriptif que le nom du fichier du VI. Cette fonction est importante pour localiser des VIs dans le cas où le titre de la fenêtre du VI doit être traduit dans une autre langue. Pour modifier le titre de la fenêtre du VI lors de son édition, sélectionnez **Configuration du VI** et **Options des fenêtres** dans le

menu déroulant du haut comme indiqué dans l'illustration suivante. Désélectionnez **Identique au nom du VI** et entrez le nouveau titre de la fenêtre du VI.



L'option **Boîte de dialogue** empêche l'utilisateur d'agir sur d'autres fenêtres tant que celle-ci reste ouverte, exactement comme une boîte de dialogue système.

Remarquez que sur un système UNIX, cet élément de menu ne vous empêche pas d'afficher les fenêtres d'autres applications au premier plan. Il est impossible de garder une fenêtre au-dessus de toutes les autres.

Autoriser les menus locaux pendant l'exécution détermine si des objets de cette face-avant peuvent disposer d'un menu local des opérations de données en mode Exécution.

Lorsque vous sélectionnez l'élément de menu **Mettre en évidence les booléens associés à la touche <Entrée>**, tous les booléens actuellement associés à la touche <Entrée> (**Windows et HP-UX**) ou <Retour> (**Macintosh et Sun**) sont mis en surbrillance. Vous pouvez associer des touches à des commandes en utilisant la boîte de dialogue Utilisation du clavier, décrite dans la section *Option d'utilisation du clavier pour les commandes* du chapitre 8, *Introduction aux objets de la face-avant*.

L'élément de menu **Ajuster à la taille de l'écran** ajuste automatiquement le panneau d'un VI à la taille de l'écran lorsque celui-ci s'exécute. Le VI n'enregistre pas sa position et sa taille d'origine. Il reste donc au même emplacement lorsqu'il revient en mode Edition.

Centrage automatique centre la face-avant sur l'écran.

Les autres éléments de menu affichent ou masquent les différentes fonctions de fenêtre. Vous pouvez masquer certains boutons en les désélectionnant ou masquer la totalité de la barre d'outils en désélectionnant l'élément de menu **Visualiser la barre d'outils**.

L'option **Activer l'enregistrement/l'impression après l'exécution** active ou désactive l'enregistrement des données automatique (reportez-vous à la section *Enregistrement des données de la face-avant* du chapitre 4, *Exécution et mise au point des VIs et sous-VIs*) et l'impression par programme (section *Impression contrôlée par programme* du chapitre 5, *Impression et documentation des VIs*).

Pour modifier le titre de la fenêtre du VI par programme, reportez-vous au chapitre 21, *VI Serveur*.

En désélectionnant Manipulation automatique des menus lors du lancement, **vous** désactivez la barre de menus d'exécution jusqu'à ce que vous soyez prêt à manipuler les sélections de menu en utilisant la fonction "Obtenir un menu". Pour plus d'informations, consultez la section *Gestion de la sélection des menus* plus loin dans ce chapitre.

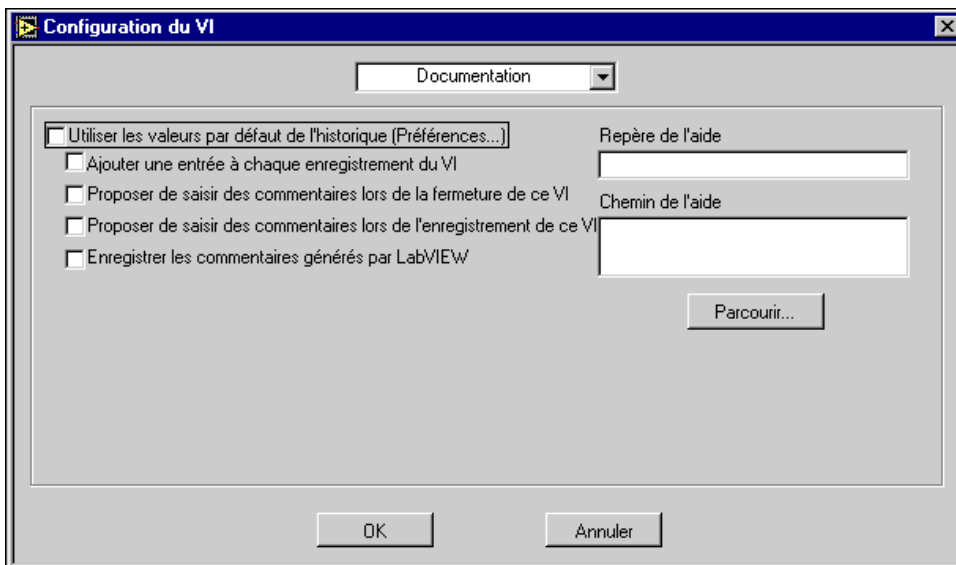


Avertissement

Si vous masquez la barre de menus et la barre d'outils, vous n'aurez plus de moyen visible pour passer du mode Exécution au mode Edition. Utilisez les touches d'accès rapide <Ctrl-m> (Windows), <commande-m> (Macintosh), <meta-m> (Sun), ou <Alt-m> (HP-UX) correspondant à l'élément de menu Exécution» Passer en mode Edition pour revenir au mode Edition du VI, mode dans lequel la barre de menus et la palette sont visibles.

Options de documentation

La boîte de dialogue de **Configuration du VI»Documentation**, présentée dans l'illustration suivante, vous liste les éléments de menu concernant les entrées effectuées dans la fenêtre Historique. Cette dernière affiche l'historique du développement du VI. Pour plus d'informations sur la fenêtre Historique, consultez la section *Fenêtre Historique du VI* du chapitre 27, *Gestion de vos applications*.



Pour pouvoir utiliser un des éléments de l'historique du menu **Documentation**, l'option **Utiliser les valeurs par défaut de l'historique** (dans la boîte de dialogue Préférences) doit être désélectionnée. Dans le cas contraire, les préférences de l'historique sont utilisées. Ces dernières sont identiques à l'exception du fait qu'elles définissent les valeurs par défaut utilisées lorsque vous créez un nouveau VI, même si les éléments du menu de la documentation disponible dans l'option **Configuration du VI** ne s'appliquent qu'au VI en cours.

Si vous cochez l'option **Ajouter une entrée à chaque enregistrement du VI**, une entrée est alors ajoutée à l'historique du VI à chaque enregistrement du VI. Si vous laissez le champ **Commentaire** de la fenêtre Historique vide, un simple en-tête est ajouté à l'historique. L'en-tête contient le numéro de révision, l'heure et la date, ainsi que le nom du VI.

Si vous cochez l'option **Proposer de saisir des commentaires lors de la fermeture de ce VI**, la fenêtre Historique s'affiche. Vous pouvez entrer un

commentaire à la fermeture d'un VI modifié depuis son chargement, même si vous avez enregistré les modifications.

Si vous cochez l'option **Proposer de saisir des commentaires lors de l'enregistrement de ce VI**, la fenêtre Historique s'affiche à l'enregistrement. Cette option est utile si vous préférez ajouter un commentaire sur vos modifications à la fin plutôt qu'au moment de l'édition. Dans le cas contraire, vous ne pourrez pas modifier l'historique du VI après avoir sélectionné **Enregistrer** et ce, jusqu'à la fin de l'enregistrement.



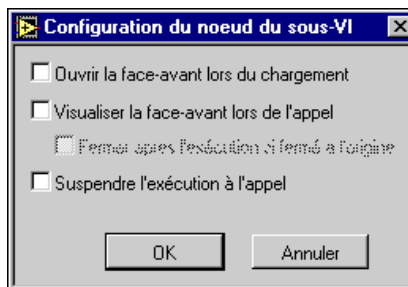
Remarque

Vous n'êtes pas invité à saisir de commentaire lorsque vous enregistrez ou fermez un VI si seul l'historique a été modifié.

Les éléments qui se situent sur la droite de la boîte de dialogue **Configuration du VI»Documentation** créent un accès aux fichiers de l'aide en ligne que vous avez créés. Positionnez votre curseur dans la zone **Repère de l'aide** et saisissez une rubrique à afficher pour ce VI. Tapez ensuite le chemin du fichier d'aide dans la zone **Chemin de l'aide**, ou cliquez sur le bouton **Parcourir...** et recherchez le fichier. Le nom du fichier et son chemin s'affichent dans la zone **Chemin de l'aide**. Une fois ces éléments de menu définis, cliquez sur l'icône de l'aide en ligne du bas de la boîte de dialogue Aide pour accéder au fichier d'aide sélectionné. Pour plus d'informations sur la création de vos propres fichiers d'aide, consultez le chapitre 5, *Impression et documentation des VIs*.

Boîte de dialogue Configuration du nœud du sous-VI

La boîte de dialogue suivante s'affiche lorsque vous sélectionnez l'option **Configuration du nœud du sous-VI...** dans le menu local du nœud d'un sous-VI du diagramme.



Ces éléments de menu sont un sous-ensemble des éléments de la boîte de dialogue Configuration du VI. La différence entre les deux réside dans le fait que vous utilisez la boîte de dialogue Configuration du sous-VI pour spécifier des éléments relatifs à un appel spécifique à un sous-VI. Toutefois, vous utiliserez les éléments d'exécution du sous-VI de la boîte de dialogue Configuration du VI afin de spécifier le comportement de tous les appels à ce VI.

Personnalisation de la barre de menus

Vous pouvez personnaliser la barre de menus pour les VIs que vous concevez. Par contre, vous ne pourrez voir les barres de menus personnalisées que lors de l'exécution du VI.

La personnalisation des menus se compose de deux parties : la création et la gestion des menus.

Vous pouvez concevoir des menus personnalisés de deux manières différentes : de manière statique pendant l'édition, ou de manière dynamique à l'exécution. Vous pouvez créer de manière statique un modèle de menu personnalisé et l'enregistrer dans un fichier appelé fichier de menu d'exécution (RTM) en utilisant l'éditeur de menus (consultez la section *Editeur de menus* de ce chapitre pour plus d'informations). L'éditeur de menus vous permet également d'associer un fichier RTM au VI en question. Lors de l'exécution du VI, il charge le menu à partir du fichier RTM associé. Vous pouvez insérer, supprimer et modifier des éléments de menu par programme, à l'exécution, à partir du diagramme en utilisant les fonctions du G (consultez la section *Fonctions de menu dynamiques* de ce chapitre pour plus d'informations).

Le diagramme gère les éléments de menus personnalisés. Chaque élément de menu possède un identificateur de chaîne de caractères unique, appelé tag, qui ignore la casse. À chaque fois que vous sélectionnez un élément de menu, vous pouvez récupérer le tag de l'élément du menu sélectionné par programme en utilisant la fonction "**Élément de menus sélectionné**" (consultez la section *Gestion de la sélection des menus* de ce chapitre pour plus d'informations). En fonction de la valeur du tag, vous pouvez utiliser un gestionnaire dans le diagramme pour chacun des éléments de menu.



Remarque

Les barres de menus personnalisées sont également appelées des menus d'exécution.

Editeur de menus

L'éditeur de menus propose une interface de création et d'édition de fichiers RTM et la possibilité de les associer au VI en question. Vous pouvez faire appel à l'éditeur de menus dans un VI en sélectionnant **Édition»Éditer un menu...** A présent, vous pouvez associer trois types de menus à un VI comme indiqué dans l'illustration 6-1 : **Par défaut**, **Minimal** et **Personnalisé**. L'option **Par défaut** présente un menu standard. L'option **Minimal** ne propose pas les éléments les moins fréquemment utilisés du menu standard. L'option **Personnalisé** vous permet d'associer un fichier RTM au VI. Les types Par défaut et Minimal ne peuvent pas être modifiés.

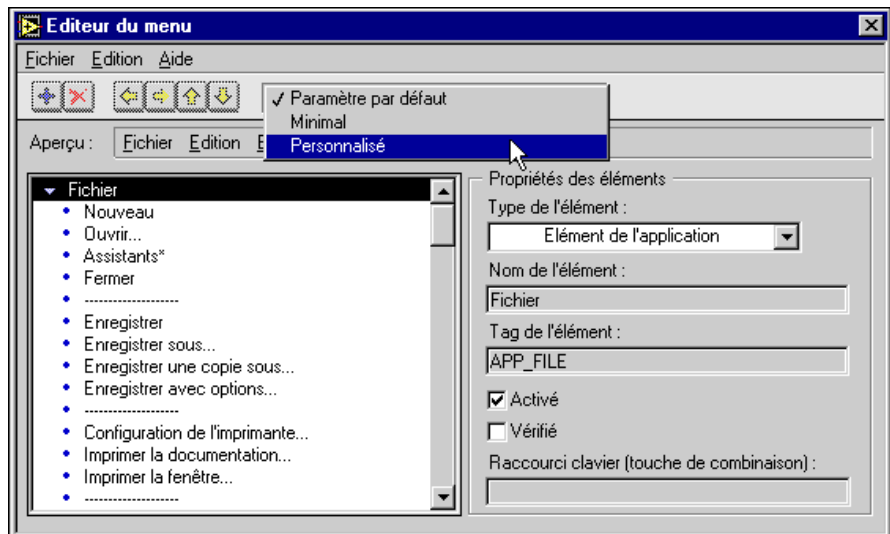


Figure 6-1. Editeur de menus



L'éditeur de menus affiche la hiérarchie du menu à gauche et les propriétés de l'élément de menu sélectionné à droite. Vous pouvez visualiser l'apparence du menu à l'exécution dans la zone **Aperçu**. Vous pouvez insérer un nouvel élément de menu en cliquant sur le bouton de la barre d'outils affichée, ou en sélectionnant les options d'insertion du menu **Édition**.

Il y a trois types d'éléments de menu : *Élément utilisateur*, *Élément d'application* ou *Séparateur*. Vous pouvez modifier le type de l'élément de menu en en choisissant un autre dans le menu déroulant **Type d'élément**.

Un **Elément utilisateur** peut être géré par programme dans le diagramme. Il porte un nom, qui est la chaîne de caractères qui s'affiche dans le menu actuel, et un tag qui est un identificateur de chaîne de caractères unique qui ignore la casse. Le tag identifie l'Elément utilisateur dans le diagramme. Pour faciliter l'édition, à chaque fois que vous entrez un nom, il est directement copié en un tag. Toutefois, vous pouvez toujours éditer le tag pour le différencier du nom. Pour qu'un élément de menu soit valide, son tag doit contenir une valeur. Les éléments de menu invalides sont signalés par trois points d'interrogation (???). L'éditeur de menus s'assure toujours que le tag est unique pour une hiérarchie de menu donnée en ajoutant des chiffres lorsque c'est utile.

Un Elément utilisateur peut être activé ou désactivé, coché ou non, en définissant les attributs respectifs. Vous pouvez définir un raccourci (accélérateur) pour un Elément utilisateur en tapant une combinaison de touches appropriée. Le raccourci comporte le format <Touche de menu[-Maj]-touche>.

**Remarque**

Les utilisateurs de PC peuvent se servir des traits de soulignement dans les noms des éléments afin de faciliter la sélection du menu par l'utilisation de la touche <ALT>. Par exemple, pour sélectionner un élément appelé Fichier/Quitter en utilisant <ALT+F+Q>, utilisez Fichier et Quitter comme noms d'éléments.

**Remarque**

La touche Maj et les touches de fonction dans les raccourcis ne s'affichent pas avec Macintosh OS 7 ou inférieur. Pour que les menus soient compatibles avec le Macintosh OS 7, évitez l'utilisation des touches Maj ou des touches de fonction dans les raccourcis.

**Remarque**

Sous une plate-forme Macintosh, tous les éléments de la barre de menus doivent être des menus déroulants. Toutefois, les utilisateurs de Windows 95/NT et d'UNIX peuvent créer des éléments de barre de menus qui peuvent être directement sélectionnés. Il vaut mieux éviter d'utiliser des éléments de barre de menus qui peuvent être directement sélectionnés sous Macintosh.

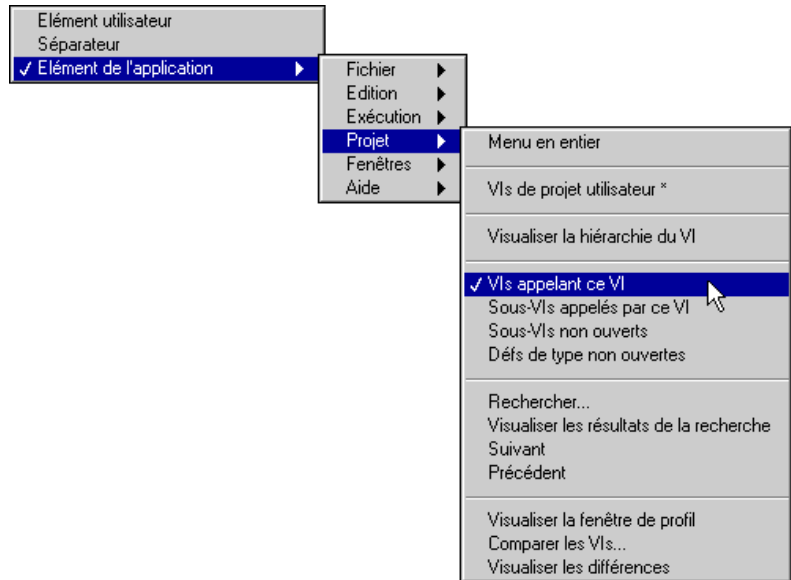


Figure 6-2. Choix d'un Elément d'application

L'**Elément d'application** est un élément fourni par le logiciel en langage G. Ce sont des éléments qui font partie du menu par défaut. Pour sélectionner un élément en particulier, choisissez **Type d'élément» Elément d'application** et suivez la hiérarchie comme indiqué dans l'illustration. Grâce à cette méthode, vous pouvez ajouter des éléments de manière individuelle ou même un ensemble de sous-menus. Les Eléments d'application sont gérés de manière implicite par le logiciel. Ces tags d'éléments ne s'affichent pas dans les diagrammes. Vous ne pouvez pas modifier le nom, le tag ou d'autres attributs d'un Elément d'application. Le logiciel réserve les tags qui commencent par APP_ pour les Eléments d'application. Les tags des Eléments d'application sont répertoriés dans le tableau 6-1 à la fin de ce chapitre.

L'élément **Séparateur** insère une ligne de séparation dans le menu. Vous ne pouvez pas définir d'attributs pour un élément Séparateur.

Vous pouvez déplacer les éléments dans la hiérarchie du menu en cliquant sur les boutons représentant des flèches dans la barre d'outils ou en utilisant les options de manipulation de la hiérarchie dans le menu **Editer**, ou encore en utilisant simplement le glisser-déposer. Les hiérarchies peuvent être réduites ou agrandies afin d'en améliorer l'affichage. Pour cela, cliquez sur les glyphes de sous-menu ou choisissez parmi les options réduire/agrandir du menu **Editer**.

Vous pouvez charger ou enregistrer les fichiers RTM en utilisant les options du menu **Fichier**. Vous pouvez utiliser les **options du menu Editer** pour couper, copier, coller, insérer et supprimer des éléments de menu (consultez la section *Options de menu de l'éditeur de menus* de ce chapitre pour plus d'informations). Vous pouvez obtenir de l'aide sur n'importe quelle commande de l'éditeur de menus en choisissant **Aide»Visualiser l'aide** et en plaçant la souris sur la commande en question.

Options de menu de l'éditeur de menus

Les options suivantes sont disponibles dans le menu **Fichier** :

Ouvrir : ouvre un fichier de menu d'exécution existant. L'ouverture d'un fichier RTM fait passer l'éditeur de menus en type Personnaliser.

Nouveau : crée un fichier de menu d'exécution. Il vous invite à enregistrer tout fichier RTM précédemment édité.

Enregistrer : enregistre le menu d'exécution en cours.

Enregistrer sous : enregistre le menu d'exécution en cours dans un autre fichier.

Fermer : quitte l'éditeur de menus.

Les options suivantes sont disponibles dans le menu **Edition** :

Couper : supprime l'élément de menu ou le texte actuellement sélectionné et le copie dans le Presse-papiers.

Copier : copie dans le Presse-papiers l'élément de menu ou le texte actuellement sélectionné.

Copier menu en entier : copie l'intégralité de la hiérarchie du menu dans le Presse-papiers.

Coller : colle le contenu du Presse-papiers.

Réduire/Agrandir : réduit ou agrandit l'élément de sous-menu actuellement sélectionné.

Réduire tout : réduit tous les éléments de sous-menu.

Agrandir tout : agrandit tous les éléments de sous-menu.

Insérer un élément utilisateur : insère un élément de menu utilisateur à la suite de l'élément de menu sélectionné.

Insérer un séparateur : insère un séparateur à la suite de l'élément de menu sélectionné.

Insérer un élément d'application : affiche un sous-menu qui permet de choisir parmi différents éléments d'application.

Effacer l'élément : supprime l'élément de menu sélectionné.

Transformer en élément principal : transforme les éléments qui suivent l'élément sélectionné en sous-éléments de l'élément sélectionné.

Transformer en sous-élément : transforme l'élément sélectionné en un élément de sous-menu du précédent élément de menu.

Déplacer l'élément vers le haut : déplace l'élément sélectionné vers le haut. Si l'élément sélectionné contient des sous-éléments, ils sont alors déplacés avec l'élément sélectionné.

Déplacer l'élément vers le bas : déplace l'élément sélectionné vers le bas. Si l'élément sélectionné contient des sous-éléments, ils sont alors déplacés avec l'élément sélectionné.

Options de la barre d'outils de l'éditeur de menus

Les options suivantes sont proposées dans la barre d'outils.



Insère un élément de menu utilisateur à la suite de l'élément de menu sélectionné.



Supprime l'élément de menu sélectionné.



Transforme les éléments qui suivent l'élément sélectionné en sous-éléments de l'élément sélectionné.



Transforme l'élément sélectionné en un élément de sous-menu de l'élément de menu précédent.



Déplace l'élément sélectionné vers le haut. Si l'élément sélectionné contient des sous-éléments, ils sont déplacés en même temps que l'élément sélectionné.



Déplace l'élément sélectionné vers le bas. Si l'élément sélectionné contient des sous-éléments, ils sont déplacés en même temps que l'élément sélectionné.

Gestion de la sélection des menus



Cette section présente une description détaillée des fonctions **Élément de menus sélectionné** et **Activer le repérage des menus**. Ces descriptions apparaissent également dans la section *Fonctions de gestion de la sélection de menu* plus loin dans ce chapitre. Les fonctions fonctionnent dans des menus identifiés par un refnum. Le refnum du menu d'un VI est obtenu grâce à la constante Menu du VI en cours, illustrée ci-contre et dans les illustrations suivantes, 6-3 et 6-4.

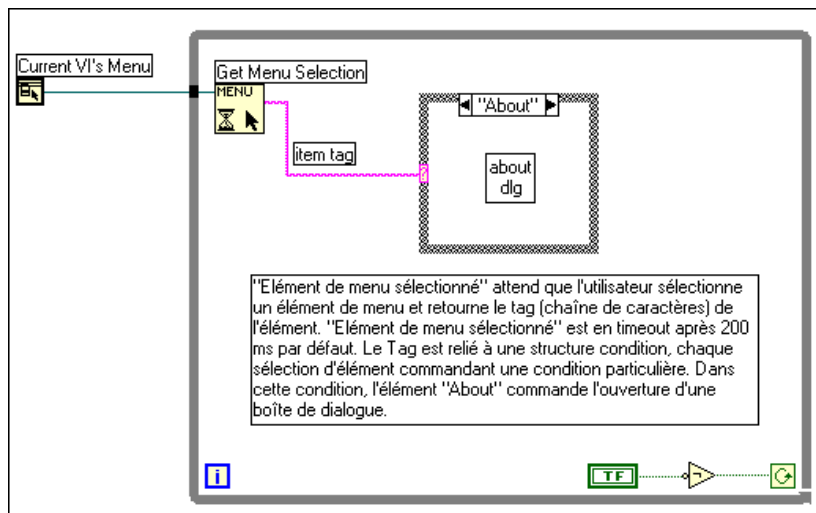


Figure 6-3. Fonction "Élément de menus sélectionné"

Une fois que vous avez sélectionné un élément, vous ne pouvez pas en sélectionner d'autres, et ce jusqu'à ce que la fonction **Élément de menus sélectionné** lise le premier élément. Toutefois, une fois qu'un élément est lu, vous pouvez sélectionner un nouvel élément de menu, même si le précédent élément n'a pas encore été traité. Cela ne fonctionne pas lorsque le délai de traitement d'une sélection est long. Dans ce cas, la fonction **Élément de menus sélectionné** est appelée en mode menu, où le repérage des menus est bloqué après la lecture de la sélection. Le menu est activé une fois que vous avez traité la sélection en utilisant la fonction **Activer le repérage des menus**, comme indiqué dans l'illustration suivante.

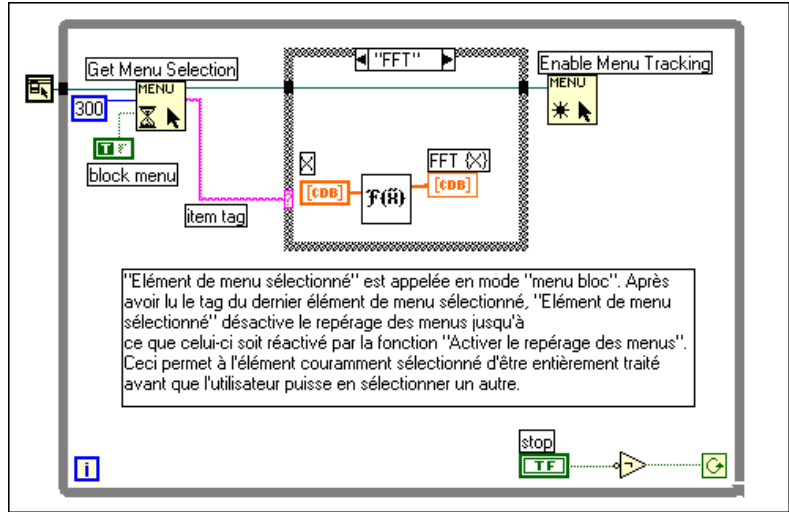


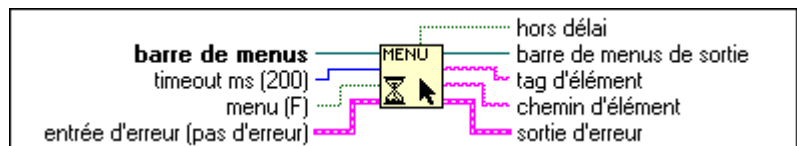
Figure 6-4. Activer le repérage des menus

Fonctions de gestion de la sélection de menu

Les fonctions de gestion de la sélection de menu suivantes sont disponibles.

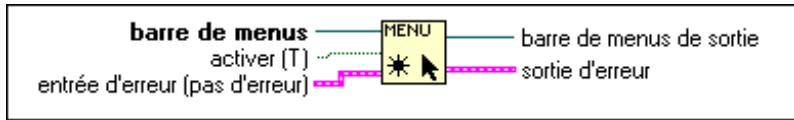
Élément de menus sélectionné

Renvoie le **Tag d'élément** du dernier élément de menus sélectionné, et attend de manière facultative le `timeout` milliseconde. Le **Chemin d'élément** est une chaîne de caractères qui décrit la position de l'élément dans la hiérarchie du menu, qui est le format d'une liste de tags de menu séparés par deux points (:). Si **Menu** est True, la sélection de menu est bloquée à la suite de la lecture d'un tag d'élément.



Activer le repérage des menus

Active ou désactive le repérage des sélections de menu.



Les fonctions s'exécutent dans des menus identifiés par un refnum. Le refnum du menu d'un VI est obtenu grâce à la constante Menu du VI en cours, illustrée ci-contre. Les éléments sont identifiés par un tag d'élément (chaîne de caractères) et parfois par un chemin d'élément (chaîne de caractères), qui est une liste des tags d'élément de la racine de l'arborescence du menu jusqu'à l'élément et séparée par deux points.

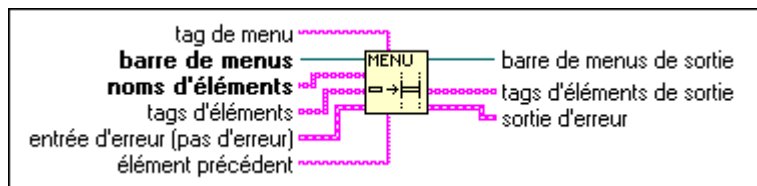
Fonctions de menu dynamiques

Pendant l'exécution d'un VI, vous pouvez modifier les éléments de la barre de menus du VI en utilisant les fonctions de gestion de menu **Eléments de menus**, **Supprimer des éléments de menus**, **Obtenir des infos sur un élément de menus**, **Définir les infos de l'élément de menus** et **Infos sur un raccourci de menus**. Vous trouverez ces fonctions dans la palette **Commande de l'application**, à laquelle vous accédez en sélectionnant **Fonctions»Commande de l'application**. Les fonctions s'exécutent dans des menus identifiés par un refnum. Le refnum de menu d'un VI est obtenu par le biais du menu de la constante du VI en cours, illustré précédemment.



Insérer des éléments de menus

Insère des éléments de menu dans une barre de menus, ou un sous-menu dans une barre de menus.



Tag de menus spécifie le sous-menu où les éléments sont insérés. Si vous ne spécifiez pas de **tag de menus**, les éléments sont insérés dans la **barre de menus**.

Nom d'élément et **tags d'éléments** identifient les éléments à insérer dans le menu. Le type de **noms d'éléments** et les **tags d'éléments** peuvent être un tableau de chaînes de caractères (pour l'insertion de plusieurs éléments) ou une simple chaîne de caractères (pour l'insertion d'un seul élément). Vous pouvez câbler soit les **noms d'éléments** soit les **tags d'éléments**, auquel cas à la fois les noms et les tags obtiennent la même valeur. S'il vous faut des noms et des tags différents pour chaque élément, vous devez alors câbler des valeurs séparées pour les **noms d'éléments** et les **tags d'éléments**.

Élément précédent spécifie la position où insérer les nouveaux éléments. **Élément précédent** peut être un tag (chaîne de caractères) d'un élément existant ou un indice de position (entier à base de zéro) dans le menu. Pour insérer un élément au début du menu, connectez **élément précédent** à un nombre inférieur à 0. Pour insérer un élément à la fin du menu, entrez un nombre supérieur au nombre d'éléments contenus dans le menu. Vous pouvez insérer des Éléments d'application en utilisant des tags d'application du tableau 6-1 à la fin de ce chapitre. Vous pouvez insérer un séparateur en utilisant le tag d'application APP_SEPARATOR. La fonction s'assure toujours que les tags de tous les éléments de menu insérés sont uniques dans la hiérarchie du menu en ajoutant un chiffre aux tags fournis, si c'est utile.

Tags d'éléments de sortie renvoie les tags réels des éléments insérés. Si **tag de menus** ou **élément précédent** (tag) n'est pas trouvé, la fonction renvoie une erreur.

L'exemple des illustrations suivantes conçoit de manière dynamique une arborescence de menus qui se compose de deux menus de haut niveau, **Fichier** et **Editer**, et d'un sous-menu **Test**.

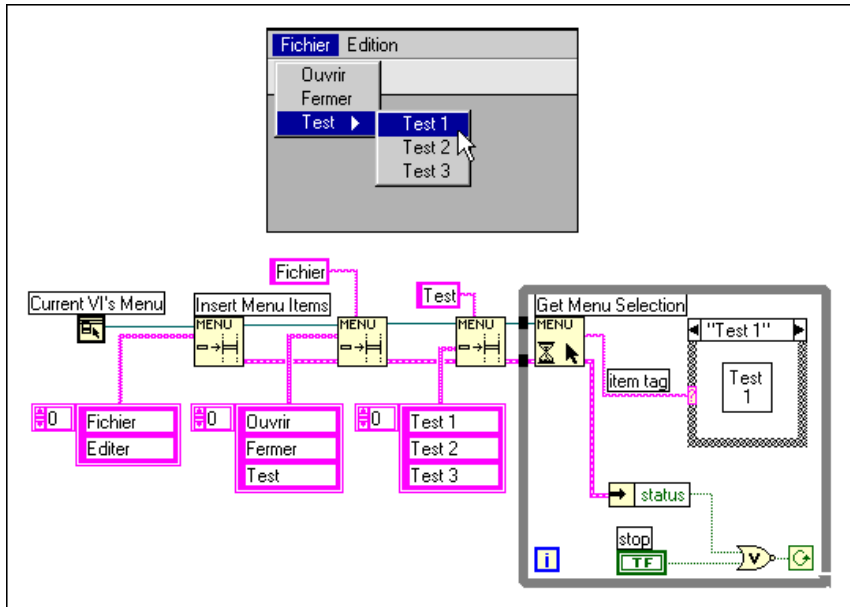
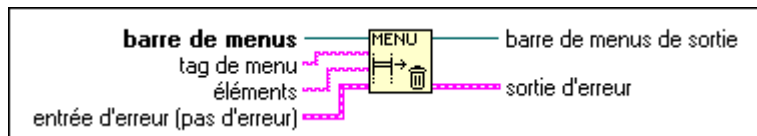


Figure 6-5. Insertion de menu dynamique

Supprimer des éléments de menus

Supprime des éléments de menus de la barre de menus ou un sous-menu dans une barre de menus.



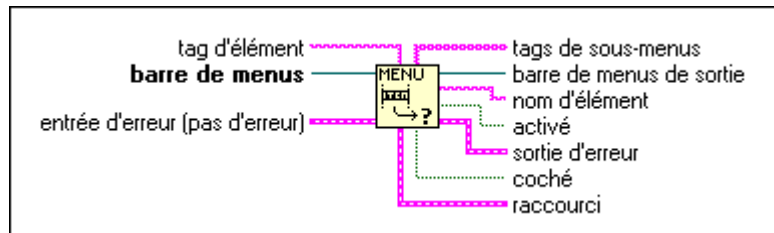
Si **tag de menus** est spécifié, les éléments sont supprimés du sous-menu spécifié par **tag de menus**. Sinon, les éléments sont supprimés de la barre de menus. La fonction renvoie une erreur si le **tag de menus** ou un des éléments spécifiés n'est pas trouvé.

Eléments peut être un tag (chaîne de caractères) d'un élément existant, un tableau de tags d'éléments existants, un indice de position (entier à base de zéro) d'un élément du menu ou un tableau d'indices de position d'éléments

dans le menu. Si vous ne câblez pas **éléments**, tous les éléments de la barre de menus seront supprimés. S'il existe un sous-menu dans un des éléments spécifiés, le sous-menu et tout son contenu sont supprimés de manière automatique. Vous pouvez supprimer des Eléments d'application en utilisant les tags d'application répertoriés dans le tableau 6-1 à la fin de ce chapitre. Etant donné que les séparateurs n'ont pas de tags uniques, il est préférable de les supprimer en utilisant leurs indices de position.

Obtenir des infos sur un élément de menus

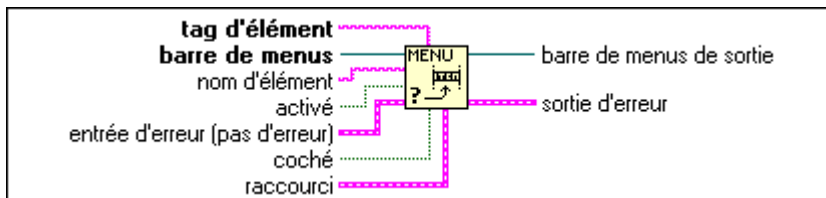
Renvoie les attributs de l'élément de menus spécifié par le tag d'élément.



Les attributs d'élément sont **nom d'élément** (la chaîne de caractères qui s'affiche dans le menu), **activé** (False désigne le fait que l'élément est grisé), **coché** (spécifie s'il y a une encoche à côté de l'élément), et **raccourci** (touche d'accès rapide). Si l'élément possède un sous-menu, ses tags d'élément sont renvoyés sous la forme d'un tableau de chaînes de caractères dans **tags de sous-menu**. Si **tag d'élément** n'est pas connecté, les éléments de la barre de menus sont renvoyés. Si **tag d'élément** n'est pas valide, une erreur est renvoyée.

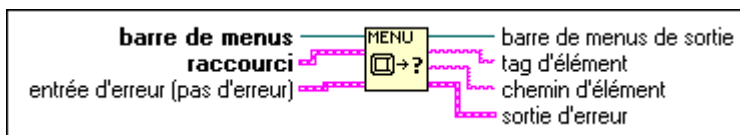
Définir les infos de l'élément de menus

Définit les attributs d'un élément de menus spécifié par le biais du tag du menu et de l'élément. Les attributs d'élément sont le nom de l'élément (la chaîne de caractères qui s'affiche dans le menu), **activé** (False désigne le fait que l'élément est grisé), **coché** (spécifie s'il y a une encoche à côté de l'élément), et **raccourci** (touche d'accès rapide). Les attributs qui ne sont pas câblés ne sont pas modifiés. Si un tag d'élément n'est pas valide, une erreur est renvoyée.



Infos sur un raccourci de menus

Renvoie l'élément de menus accessible grâce à un raccourci donné.



Chemin d'élément est une chaîne de caractères de tags d'élément de menus séparés par deux points (:).

Raccourci est une chaîne de caractères (touche) et un Booléen (spécifie si la touche Maj est incluse ou non).

Tags d'Elément d'application

Vous pouvez insérer des éléments d'application en utilisant les tags d'Elément d'application répertoriés dans le tableau suivant.

Tableau 6-1. Tags d'Elément d'application

Elément d'application	Tag d'élément
Fichier	APP_FILE
Nouveau	APP_NEW
Ouvrir...	APP_OPEN
Fermer	APP_CLOSE
Enregistrer	APP_SAVE
Enregistrer sous...	APP_SAVEAS
Enregistrer une copie sous...	APP_SAVE_A_COPY_AS
Enregistrer avec options...	APP_SAVE_WITH_OPTIONS

Tableau 6-1. Tags d'Élément d'application (Suite)

Élément d'application	Tag d'élément
Configuration de l'imprimante...	APP_PRINTER_SETUP
Imprimer la documentation...	APP_PRINT_DOCUMENTATION
Imprimer la fenêtre...	APP_PRINT_WINDOW
Editer une bibliothèque de VIs...	APP_EDIT_VI_LIBRARY
Editer un modèle...	APP_EDIT_TEMPLATE
Compilation globale...	APP_MASS_COMPILE
Convertir un driver d'instrument CVI...	APP_CONVERT_CVI
Quitter	APP_EXIT
Editer	APP_EDIT
Annuler	APP_UNDO
Revenir	APP_REDO
Couper	APP_CUT
Copier	APP_COPY
Coller	APP_PASTE
Effacer	APP_CLEAR
Importer un fichier image...	APP_IMPORT_PICT_FROM_FILE
Préférences...	APP_PREFERENCES
Nom utilisateur...	APP_USER_NAME
Effacer le cache de mot de passe	APP_CLEAR_PASSWORD_CACHE
Sélectionner le type de palette	APP_SELECT_PALETTE_SET
Editer les palettes...	APP_EDIT_PALETTES
Exécuter	APP_OPERATE
Stop	APP_STOP
Imprimer à la fin de l'exécution	APP_PRINT_AT_COMPLETION

Tableau 6-1. Tags d'Élément d'application (Suite)

Élément d'application	Tag d'élément
Enregistrer à la fin de l'exécution	APP_LOG_AT_COMPLETION
Enregistrement des données	APP_DATA_LOGGING
Enregistrer...	APP_LOG
Récupérer...	APP_RETRIEVE
Purger les données...	APP_PURGE_DATA
Modifier la liaison au fichier d'enregistrement...	APP_CHANGE_LOG_BINDING
Effacer la liaison du fichier d'enregistrement	APP_CLEAR_LOG_BINDING
Interrompre à l'appel	APP_SUSPEND_WHEN_CALLED
Réinitialiser tout par défaut	APP_REINIT_ALL_TO_DEFAULT
Projet	APP_PROJECT
VIs de projet utilisateur *	APP_USER_PROJECT_VIS
Visualiser la hiérarchie des VIs	APP_SHOW_VI_HIER
VIs appelant ce VI	APP_THIS_VIS_CALLERS
Sous-VIs appelés par ce VI	APP_THIS_VIS_SUBVIS
Sous-VIs non ouverts	APP_UNOPENED_SUBVIS
Défs. de type non ouvertes	APP_UNOPENED_TYPEDEFS
Rechercher...	APP_FIND
Afficher les résultats de la recherche	APP_SHOW_SRCH_RSLTS
Suivant	APP_FIND_NEXT
Précédent	APP_FIND_PREV
Fenêtre d'optimisation	APP_SHOW_PROFILE_WINDOW
Comparaison des VIs...	APP_COMPARE_VIS
Visualiser les différences	APP_SHOW_DIFFERENCES

Tableau 6-1. Tags d'Élément d'application (Suite)

Élément d'application	Tag d'élément
Exporter les chaînes de caractères...	APP_EXPORT_STRINGS
Importer les chaînes de caractères...	APP_IMPORT_STRINGS
Fenêtres	APP_WINDOWS
Face-avant	APP_SHOW_PANEL_DIAGRAM
Infos sur le VI...	APP_SHOW_VI_INFO
Historique	APP_SHOW_HISTORY
Palette de fonctions	APP_SHOW_CTRLN_FCTNS
Palette d'outils	APP_SHOW_TOOLS
Presse-papiers	APP_SHOW_CLIPBOARD
Liste des erreurs	APP_SHOW_ERRORS
Mosaïque verticale	APP_TILE_LEFT_RIGHT
Mosaïque horizontale	APP_TILE_UP_DOWN
Taille normale	APP_FULL_SIZE
Ouvrir les fenêtres *	APP_OPEN_WINDOWS
Aide	APP_HELP
Visualiser l'aide	APP_SHOW_HELP
Verrouiller l'aide	APP_LOCK_HELP
Aide simple	APP_SIMPLE_HELP
Référence en ligne...	APP_ONLINE_REF
Aide en ligne du VI	APP_ONLINE_HELP_FOR_VI
Fichiers d'aide *	APP_HELP_FILES
A propos de LabVIEW...	APP_ABOUT

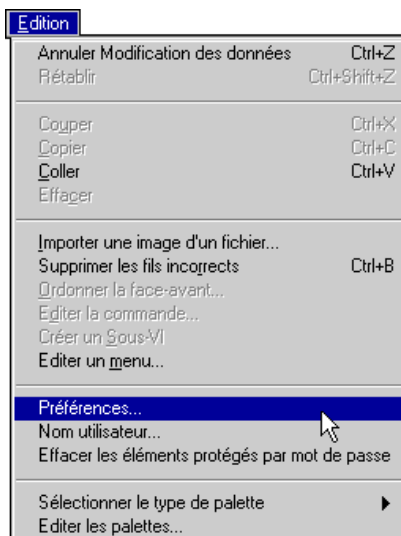
Personnalisation de votre environnement

Ce chapitre traite de la manière de personnaliser votre environnement en paramétrant les préférences de l'éditeur pour des fonctions telles que l'impression, l'affichage et l'option **Annuler**, et en modifiant le contenu des palettes **Commandes** et **Fonctions**.

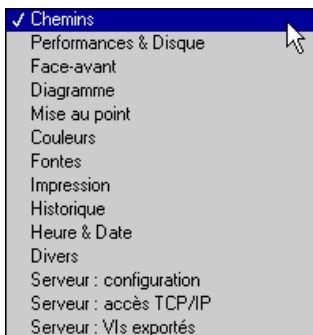
Paramètres des préférences

Pour personnaliser votre application et configurer l'éditeur, utilisez la boîte de dialogue Préférences.

Pour accéder à la boîte de dialogue Préférences, choisissez l'option **Edition»Préférences...**, comme indiqué dans l'illustration suivante.

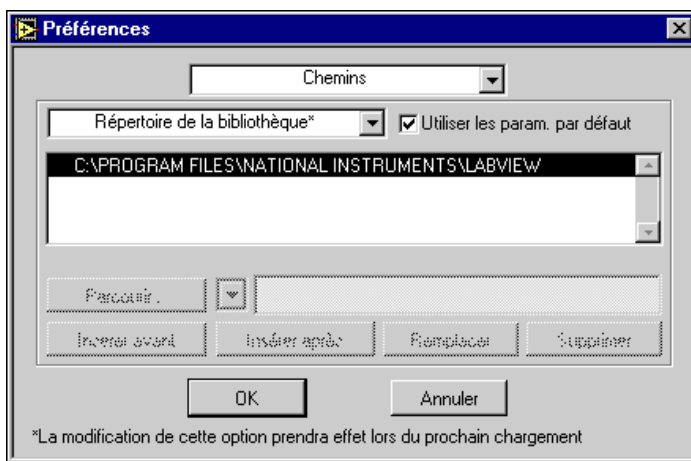


Utilisez le menu déroulant du haut de la boîte de dialogue pour sélectionner une catégorie de préférences, comme indiqué dans l'illustration suivante.



Préférences des chemins

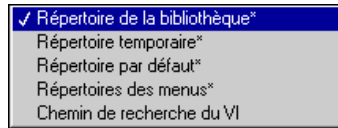
Vous pouvez spécifier les répertoires de recherche des VIs ainsi que les chemins utilisés pour les fichiers temporaires et le répertoire de la bibliothèque. Pour cela, sélectionnez **Chemins** dans le menu déroulant de la boîte de dialogue Préférences.



Remarque

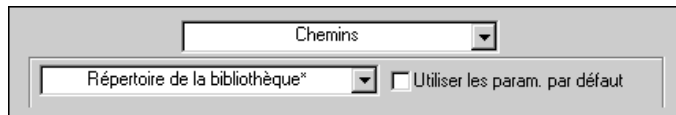
Le format des noms de chemin (les deux points, les barres obliques ou les barres obliques inverses) diffère légèrement suivant les plates-formes.

Dans la boîte de dialogue Préférences des chemins se trouve un autre menu déroulant (illustration suivante), à partir duquel vous pouvez sélectionner la catégorie de chemin que vous voulez visualiser ou modifier.



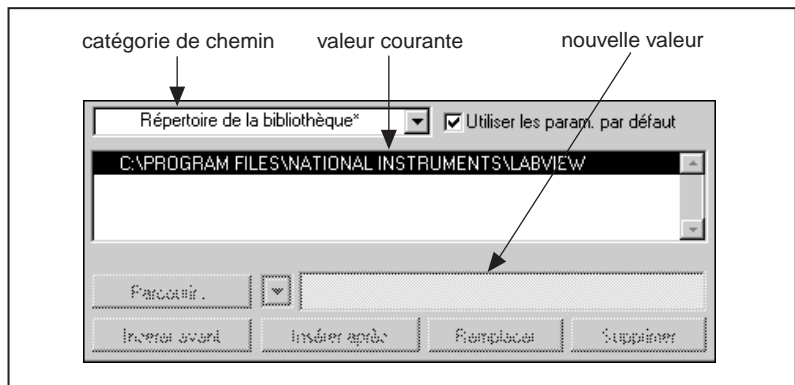
Dans la boîte de dialogue qui s’affiche, plusieurs éléments sont grisés et inactifs, le G étant configuré de manière à utiliser le chemin par défaut.

Pour modifier une de ces préférences, désélectionnez la case **Utiliser les paramètres par défaut**, comme indiqué dans l’illustration suivante.



Répertoires de la bibliothèque, répertoire temporaire, répertoire par défaut et répertoire des menus

Le répertoires de la bibliothèque, le répertoire temporaire, le répertoire par défaut et le répertoire des menus sont des répertoires uniques (dossiers). Lorsque vous éditez un de ces chemins, vous pouvez taper un nouveau chemin et le remplacer, ou parcourir la boîte de dialogue de fichiers pour en sélectionner un autre. L’illustration suivante vous montre comment faire.



Le **Répertoire de la bibliothèque** affiche le nom de chemin absolu du répertoire qui contient `vi.lib` et de tous les répertoires des bibliothèques que vous rajoutez. C'est, par défaut, le répertoire qui contient votre environnement de développement en G.

Le **Répertoire temporaire** affiche le chemin absolu du répertoire des fichiers temporaires. Ce dernier varie en fonction de la plate-forme de la manière suivante :

- **(Windows)** La valeur par défaut est le répertoire qui contient votre environnement de développement en G.
- **(Macintosh)** Sous le Système 7, le répertoire par défaut est un dossier invisible appelé `Temporary Items` au début de votre disque dur. Apple recommande cet emplacement pour le stockage des fichiers temporaires. Si votre environnement de développement se bloque, les fichiers temporaires sont déplacés dans la corbeille au redémarrage.
- **(UNIX)** Le répertoire par défaut est `/tmp`.

Le **Répertoire par défaut** affiche le chemin absolu du répertoire par défaut qui est celui que la boîte de dialogue des fichiers affiche au départ. La fonction **Répertoire par défaut** de la palette **Fonctions»E/S de fichiers» Constantes de fichiers** retourne également cette valeur. La valeur par défaut est le répertoire de travail actuel. Les modifications apportées au répertoire par défaut prennent immédiatement effet.

Le **Répertoire des menus** affiche le chemin du répertoire qui contient les fichiers des menus palette qui décrivent l'organisation des palettes **Commandes** et **Fonctions**. La valeur par défaut est le **Répertoire des menus** dans votre **Répertoire de la bibliothèque**. En général, il n'est pas nécessaire de le modifier. Par contre, ce sera le cas si vous vous trouvez sur un réseau et que chaque utilisateur nécessite son jeu de menus personnalisé.



Remarque

Toutes les modifications apportées aux options de Répertoire de la bibliothèque, de Répertoire temporaire, de Répertoire de menus et de Répertoire par défaut ne prennent effet qu'au redémarrage de votre environnement de développement en G.

Chemin de recherche du VI

Le **Chemin de recherche du VI** n'est utilisé que lorsque votre environnement de développement en G recherche un sous-VI, une commande, ou un sous-programme externe absent de l'emplacement prévu. Il vous permet de lister le chemin des répertoires dans lesquels le G effectue la recherche. Lorsque vous éditez le chemin de recherche, vous avez accès à plus d'options que pour les autres chemins — vous pouvez

ajouter de nouveaux éléments à des emplacements spécifiques, supprimer des chemins, et en sélectionner parmi une liste de chemins *spéciaux*.



La liste contient les chemins recherchés par votre environnement de développement en G, dans l'ordre suivant lequel il les recherche. Pour ajouter un nouveau répertoire, choisissez d'abord à quel moment votre environnement de développement doit rechercher dans ce répertoire plutôt que dans les autres. Sélectionnez un répertoire adjacent dans la liste. Utilisez ensuite soit le bouton **Parcourir...**, qui affiche une boîte de dialogue de fichiers, soit le menu déroulant spécial qui contient une liste de chemins spéciaux, pour sélectionner le répertoire dans lequel votre environnement de développement en G doit effectuer la recherche. Modifiez ou entrez ce chemin dans la commande de chaîne de caractères située à côté de ces options. Enfin, utilisez **Insérer avant**, **Insérer après** ou **Remplacer** pour l'ajouter à la liste.

Lorsque vous sélectionnez un chemin, le G recherche normalement dans ce répertoire, mais pas dans ses sous-répertoires. Vous pouvez rendre la recherche hiérarchique en ajoutant un * comme nouvel élément de chemin.

Quelques exemples sur différentes plates-formes :

- **(Windows)** Pour rechercher dans le répertoire C:\VIS\ de manière récursive, entrez C:\VIS*
- **(Macintosh)** Pour rechercher de manière récursive dans le dossier VIS du disque dur, entrez HD:VIS:*
- **(UNIX)** Pour rechercher de manière récursive dans /usr/home/gregg/vis, entrez /usr/home/gregg/vis/*.

Utilisez le menu déroulant spécial présenté dans l'illustration suivante, situé à droite du bouton **Parcourir...** pour sélectionner plusieurs répertoires spéciaux.



Ces répertoires sont les suivants :

- <vilib> comme entrée de chemin est le chemin symbolique qui se rapporte au répertoire vi.lib à l'intérieur du **Répertoire de la bibliothèque**.
- <topvi> se rapporte au répertoire qui contient le VI principal ouvert.
- <foundvi> se rapporte à la liste des répertoires que LabVIEW construit à chaque chargement du VI. Pendant la recherche, tout répertoire dans lequel LabVIEW trouve un sous-VI, ou tout répertoire que vous sélectionnez manuellement, est ajouté à cette liste. Utilisez ce chemin symbolique de telle manière que si vous déplacez ou renommez un répertoire de VIs, et ensuite ouvrez un VI appelant, vous ne trouviez ce répertoire qu'une seule fois pour ce chargement.

Supprimez un chemin en utilisant le bouton **Supprimer** qui place le chemin supprimé dans la zone de chaîne de caractères, au cas où vous décideriez de le réinsérer ailleurs dans la liste.



Remarque

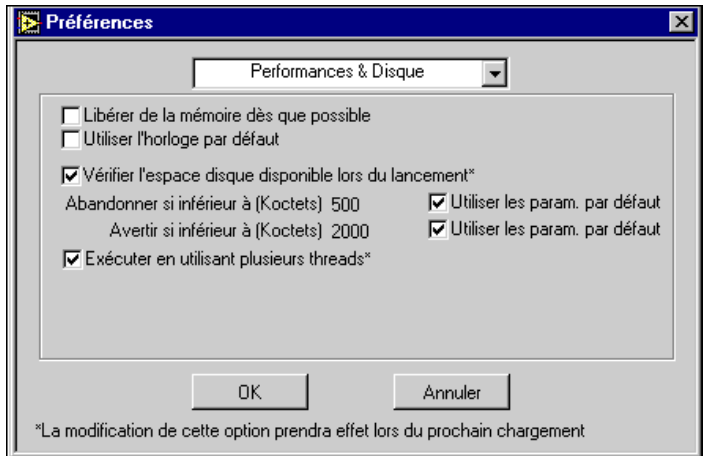
Les modifications apportées à l'option Chemin de recherche du VI ne prennent effet qu'au redémarrage de l'application.

Préférences concernant le disque et les performances

Certaines opérations d'édition nécessitent l'utilisation de fichiers temporaires sur le disque. Par exemple, lorsque vous enregistrez un VI, il est d'abord enregistré dans un fichier temporaire. Si l'enregistrement réussit, le nouveau fichier remplace alors l'original. Cette opération peut occuper une bonne partie de l'espace disque. Pour éviter les problèmes, l'espace disque disponible dans le répertoire temporaire est vérifié au moment du démarrage. Si vous avez moins de 500 Ko d'espace disque disponible, un message vous en avertit et le chargement est interrompu. Si vous en avez moins de 2 Mo, un message vous en avertit, mais le chargement se poursuit. Vous pouvez désactiver ces avertissements ou

modifier les niveaux d'alerte en utilisant la boîte de dialogue Performances & Disque à laquelle vous pouvez accéder à partir du menu déroulant Préférences.

La boîte de dialogue des préférences concernant les performances et le disque vous est présentée dans l'illustration suivante. Remarquez qu'un seul élément de menu est disponible sous Windows 3.1, et que quelques autres options le sont uniquement sur Macintosh.



Les options de cette boîte de dialogue sont les suivantes :

Libérer la mémoire dès que possible : libère la mémoire de chaque VI une fois l'exécution terminée. Cette opération peut améliorer l'utilisation de la mémoire dans certaines applications car les sous-VI libèrent leur mémoire dès la fin de l'exécution. Toutefois, elle ralentit les performances, le G devant allouer et libérer de la mémoire plus fréquemment, ce qui dans certains cas entraîne une fragmentation excessive de la mémoire.

(Windows 3.1) Utiliser l'horloge par défaut : les fonctions de séquençement d'exécution du G utilisent l'horloge intégrée de Windows. Cette horloge a, par défaut, une résolution de 55 ms (18 impulsions par seconde). Cela signifie que les VI Compteur d'impulsions d'horloge, Attendre, et Attendre un multiple de ms du G n'ont qu'une résolution de 55 ms. Remarquez que vous pouvez spécifier des résolutions d'horloge plus fines aux VI DAQ car ils utilisent les horloges intégrées des cartes DAQ pour contrôler la fréquence d'acquisition.

Augmentez la résolution des VIs Compteur d'impulsions d'horloge, Attendre, et Attendre un multiple de ms à 1 ms en augmentant la résolution de l'horloge intégrée de Windows à 1 ms. Si vous augmentez la résolution de l'horloge, le séquençement des boucles de votre logiciel sera plus précis.

Dans certains cas, vous ne pouvez pas augmenter la résolution de l'horloge au-delà de 55 ms. La modification de la résolution augmente le nombre des impulsions d'horloge de 18 interruptions à 1,000 interruptions par seconde. Si vous réalisez d'autres opérations utilisant des interruptions de façon intensive, vous dépasserez peut-être la capacité de votre PC à gérer le chargement des interruptions ce qui risquerait de le bloquer.

Les opérations DAQ utilisant des E/S programmées au lieu des DMA pour transférer des données entre la carte DAQ et la mémoire du PC utilisent intensément les interruptions. Ne modifiez pas la résolution des impulsions d'horloge à 1 ms si vous prévoyez d'utiliser une des opérations suivantes à un fort taux de transfert :

- Entrée analogique bufférisée utilisant une PC-LPM-16
- Entrée analogique bufférisée utilisant une AT-MIO-16, MC-MIO-16 ou les cartes de série Lab
- E/S numériques bufférisées utilisant une DIO-24, DIO-96 ou les cartes de série Lab
- Entrée ou sortie analogiques bufférisées avec le DMA désactivé dans WDAQCONF utilisant n'importe quelle carte DAQ

Les vitesses de transfert qui atteignent approximativement 30 K-échantillons/sec ou plus sur une machine 386 25 MHz provoquent des problèmes de chargement des interruptions avec une résolution d'horloge de 1 ms. Les vitesses de transfert qui atteignent approximativement 75 à 80 K-échantillons/sec ou plus sur une machine 486 33 MHz, ou 100 K-échantillons/sec ou plus sur une machine 486 50 MHz, peuvent également provoquer des problèmes de chargement des interruptions.

Les fonctions de séquençement des exécutions du G pour Windows 95/NT ont une résolution d'horloge de 1 ms, qui est la résolution maximale sous Windows 95/NT. Des machines plus lentes peuvent présenter des résolutions inférieures.

(Macintosh) Compacter la mémoire pendant l'exécution : indique au G de compacter la mémoire approximativement toutes les 30 secondes. Le compactage réduit la fragmentation en déplaçant la mémoire allouée vers un emplacement unique. L'inconvénient est que ce compactage prend du temps et peut provoquer de courts arrêts lors du compactage de la mémoire.

(Macintosh) Niveau de coopération : avec cette option, vous pouvez contrôler l'importance de la coopération du G avec d'autres applications en arrière-plan. Si vous le définissez **Bas**, le G ne partagera pas souvent l'horloge avec les autres applications. Le définir à un niveau bas améliore les performances du logiciel, mais ralentit celles des autres applications. Si vous le définissez **Haut**, les performances du G diminuent, mais les autres applications disposent de plus de temps pour s'exécuter.

Vérifier l'espace disque disponible au démarrage : vérifie la quantité d'espace disque disponible du répertoire temporaire lors du démarrage du logiciel. Vous pouvez décider si le G doit utiliser le répertoire par défaut dans deux cas : interrompre si inférieur à 500 Ko, ou avertir si inférieur à 2 000 Ko.

Exécuter en utilisant le multithreading : cette option n'est disponible que sur des systèmes d'exploitation qui supportent le multithreading. Si vous la désactivez, le système d'exécution se comporte comme si vous n'aviez qu'un thread unique, l'interaction de l'interface utilisateur et l'exécution des VIs ayant lieu dans le même thread. Vous pouvez avoir à utiliser cette option pour des raisons de compatibilité si vous avez des VIs qui ne se comportent pas correctement dans le système d'exécution multithreading.

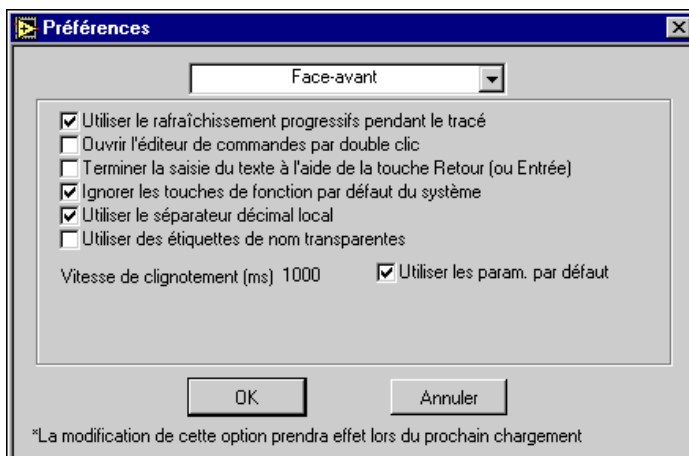


Remarque

Les modifications apportées aux options Vérifier l'espace disque disponible au démarrage, Utiliser l'horloge par défaut et Exécuter en utilisant le multithreading ne prennent effet qu'au redémarrage du logiciel. Les modifications apportées aux options Compacter la mémoire pendant l'exécution, Libérer la mémoire dès que possible et Niveau de coopération prennent effet immédiatement.

Préférences de la face-avant

La boîte de dialogue de Préférences Face-avant est présentée dans l'illustration suivante.



Les options de cette boîte de dialogue sont les suivantes :

Ouvrir l'éditeur de commandes par double clic : facilite l'accès à la fenêtre de l'éditeur de commandes où vous pouvez personnaliser l'apparence d'une commande ou d'un indicateur de face-avant.

Terminer la saisie du texte à l'aide de la touche Retour (ou Entrée) : permet d'utiliser la touche d'entrée du clavier ou du pavé numérique pour terminer la saisie du texte. Si vous sélectionnez cette option, vous pourrez incorporer de nouvelles lignes en appuyant sur <Ctrl-Entrée> (**Windows**), <option-retour> (**Macintosh**), <meta-Retour> (**Sun**) ou <Alt-Retour> (**HP-UX**).

(Windows, Macintosh) Ignorer les touches de fonction par défaut du système : par défaut, votre système d'exploitation pourrait se servir de certaines touches de fonction pour des opérations système. Par exemple, sur un PC, le fait d'appuyer sur la touche F10 revient à utiliser la touche <Arrêt>. Le Macintosh se sert généralement des touches F1 à F4 pour **Couper, Copier, Coller** et **Effacer**. Si vous activez cette option, les touches de fonction ne seront plus utilisées par le système mais par le logiciel comme touches de fonction standard.

Séparateur décimal local : utilise le séparateur décimal pour le système au lieu du point. Par exemple, dans de nombreux pays, la virgule est utilisée comme séparateur décimal. Activez cette option si vous voulez que le

logiciel agisse en fonction de la configuration de votre système. Désactivez cette option si vous voulez que le point soit toujours utilisé comme séparateur décimal.

Utiliser le rafraîchissement progressif pendant le tracé : lorsque le G met à jour une commande sans rafraîchissement progressif, il efface le contenu de la commande et redessine la nouvelle valeur. Par conséquent, l’affichage peut vaciller sensiblement. En utilisant le rafraîchissement progressif, le logiciel dessine les données dans un buffer hors écran et copie ensuite cette image à l’écran au lieu d’en effacer une partie. Vous évitez ainsi le vacillement provoqué par l’effacement et le dessin. Toutefois, cette opération peut diminuer les performances car elle nécessite plus de mémoire pour la gestion du buffer de dessin hors écran.

Utiliser des étiquettes de nom transparentes : le logiciel utilise alors des étiquettes au travers desquelles on peut distinguer les éléments placés au-dessous.

(Sun) Prise en charge du pavé numérique sur les claviers Sun : active la prise en charge du clavier Sun, y compris les pavés numériques, les touches de direction et la touche Aide. N’activez pas cette option si vous n’utilisez pas de clavier Sun (par exemple, si vous utilisez un terminal X ou un PC qui exécute un logiciel X).

Vitesse de clignotement : définit la vitesse de clignotement des objets de la face-avant à la valeur par défaut de 1 000 millisecondes ou toute autre vitesse que vous entrez. Le clignotement est un attribut de base activé par les attributs nodes. Pour plus d’informations, reportez-vous au chapitre 22, *Attribute Nodes*.

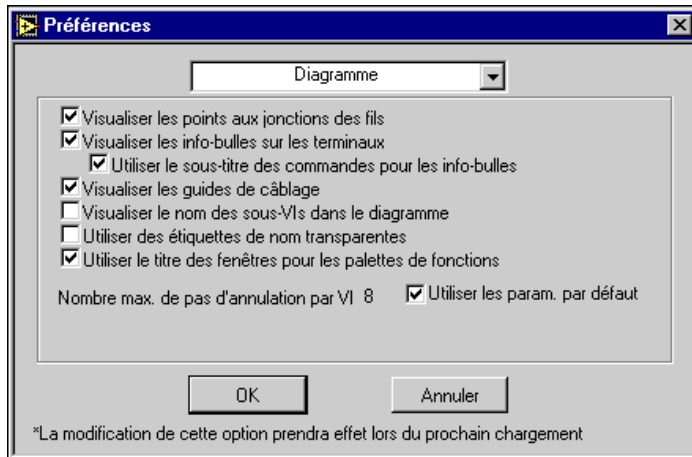


Remarque

Les modifications apportées aux options de la boîte de dialogue de Préférences Face-avant prennent effet immédiatement.

Préférences pour le diagramme

La boîte de dialogue Préférences Diagramme vous est présentée dans l'illustration suivante.



Les options de cette boîte de dialogue sont les suivantes :

Visualiser les points aux jonctions des fils : place des points aux endroits où un fil prend plusieurs directions. Cela facilite la différenciation entre les jonctions et les croisements de fils.

Visualiser les info-bulles sur les terminaux : place le nom des paramètres sous les terminaux dans les fonctions et les sous-VIs lorsque vous positionnez votre curseur au-dessus.

Utiliser un sous-titre de commande pour les info-bulles : utilise les sous-titres des commandes au lieu des étiquettes lors de l'affichage de l'aide des VIs ou lors de l'affichage des info-bulles pour les terminaux VI.

Visualiser les guides de câblage : affiche les mauvaises connexions indiquant le type de données et leur direction pour chaque paramètre lorsque le curseur est positionné au-dessus d'un nœud du diagramme.

Visualiser le nom des sous-VIs dans le diagramme : demande au logiciel d'étiqueter un sous-VI avec le nom du VI original lorsque vous le placez dans le diagramme, et d'afficher l'étiquette.

Utiliser des étiquettes de nom transparentes : le logiciel utilise alors des étiquettes au travers desquelles on peut distinguer des éléments placés au-dessus.

Utiliser les titres des fenêtres pour les palettes de fonctions : utilise le titre de fenêtre du VI au lieu du nom du VI lors de l'affichage du nom d'un VI dans les palettes de fonctions.

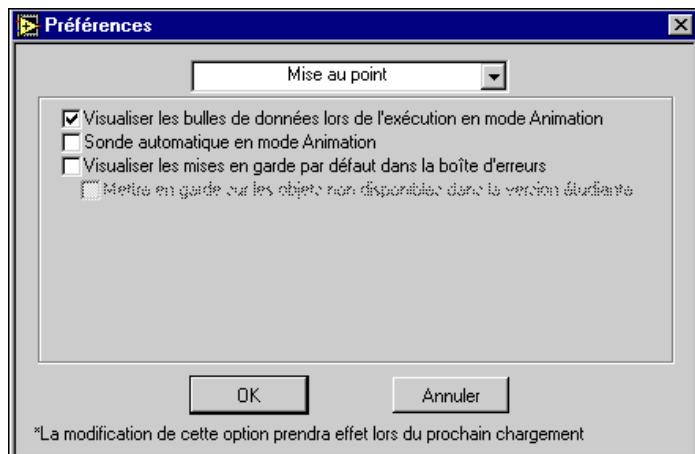
Nombre maximal d'étapes d'annulations par VI : définit le nombre d'actions que vous pouvez annuler et rétablir. Si le nombre d'actions est zéro, l'option Annuler du menu Editer est désactivée. Ce nombre est propre à chaque VI. Par conséquent, le nombre que vous tapez concerne chaque VI. Vous conserverez des ressources mémoire en définissant un petit nombre. Pour plus d'informations, consultez la section *Annuler* de ce chapitre.

 **Remarque**

Les modifications apportées aux options de la boîte de dialogue de Préférences Diagramme prennent effet immédiatement.

Préférences concernant la mise au point

La boîte de dialogue de Préférences Mise au point vous est présentée dans l'illustration suivante.



Les options de cette boîte de dialogue sont les suivantes :

Visualiser les bulles de données lors de l'exécution en mode

Animation : anime l'exécution en dessinant des bulles le long des fils pour représenter le flux des données. Désactivez cette fonction pour une mise au point plus rapide avec moins d'animation.

Sonde automatique en mode Animation : sonde automatiquement les valeurs scalaires en représentant leurs valeurs sur le diagramme. Vous pouvez désactiver cette fonction pour ne pas encombrer l'affichage.

Visualiser les mises en garde par défaut dans la boîte d'erreurs : affiche les mises en garde en plus des erreurs dans la boîte de dialogue Liste des erreurs. Une mise en garde ne signifie pas que le VI est incorrect ; elle met simplement en évidence un problème éventuel dans votre diagramme.

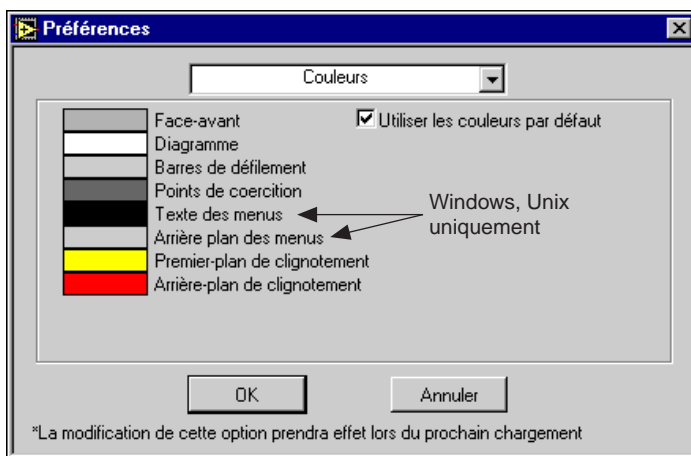
Mettre en garde sur des objets non disponibles dans la version étudiant : si vous sélectionnez l'élément de menu précédent concernant les mises en garde, cet élément inclut des mises en garde concernant les objets qui ne sont pas disponibles dans la version étudiante de LabVIEW.

 **Remarque**

Les modifications apportées aux options de la boîte de dialogue de Préférences Mise au point prennent effet immédiatement.

Préférences des couleurs

La boîte de dialogue de Préférences Couleurs vous est présentée dans l'illustration suivante.



Cette boîte de dialogue vous permet de modifier les couleurs utilisées par le logiciel pour les différents éléments. Si vous ne cochez pas la case **Utiliser les couleurs par défaut**, vous pourrez alors cliquer sur n'importe quel rectangle pour en modifier la couleur. Les options sont les suivantes :

Face-avant : sélectionnez une couleur pour la face-avant des nouveaux VIs. Cette modification n'affecte pas les VIs existants.

Diagramme : sélectionnez une couleur pour le diagramme des nouveaux VIs. Cette modification n'affecte pas les VIs existants.

Barre de défilement : sélectionnez une couleur pour les barres de défilement. Cette modification n'affecte que les VIs actuellement ouverts.

Points de coercition : sélectionnez une couleur pour les points indiquant la conversion des données au format du terminal. Cette modification n'affecte que les VIs actuellement ouverts.

(Windows, UNIX) Texte des menus : sélectionnez une couleur pour le texte utilisé dans les menus.

(Windows, UNIX) Arrière-plan des menus : vous permet de sélectionner une couleur pour l'arrière-plan utilisé dans les menus.

Premier plan de clignotement : vous permet de sélectionner la couleur du premier plan d'un objet clignotant. Cette modification n'affecte l'objet que lorsqu'il clignote. Le clignotement est un attribut de base activé avec les *attribute nodes*. Pour plus d'informations, consultez le chapitre 22, *Attribute Nodes*.

Arrière-plan de clignotement : vous permet de sélectionner la couleur de l'arrière-plan d'un objet clignotant. Cette modification n'affecte l'objet que lorsqu'il clignote.

Utiliser les couleurs par défaut : utilise le choix des couleurs par défaut pour tous les éléments répertoriés dans cette boîte de dialogue. Vous devez désactiver cette option si vous souhaitez modifier une de ces couleurs.

(UNIX) Fournir des couleurs supplémentaires : bascule entre les palettes 216 couleurs et 125 couleurs. Une plus grande palette peut provoquer un flash sensible lorsque vous basculez entre plusieurs fenêtres si votre système ne supporte que 256 couleurs.

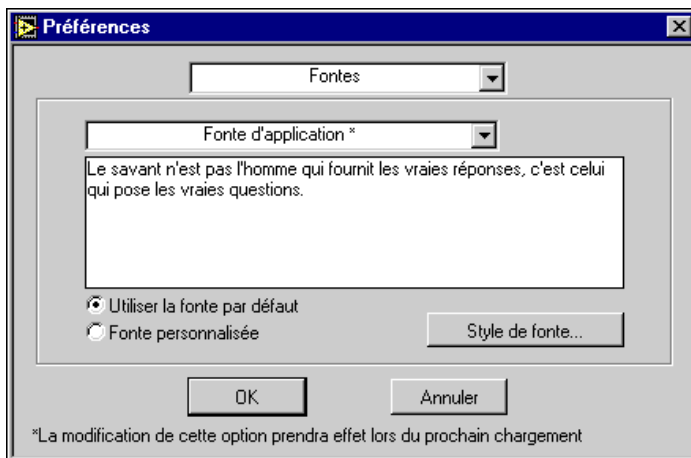


Remarque

Les modifications apportées aux options de la boîte de dialogue de Préférences Couleurs prennent effet immédiatement.

Préférences de fonte

La boîte de dialogue de Préférences Fonte vous permet de modifier trois catégories de fontes prédéfinies : la Fonte de l'application, la Fonte système et la Fonte des boîtes de dialogue. Vous pouvez sélectionner la catégorie de fonte que vous voulez modifier à partir du menu local situé au-dessus de la zone de texte. La boîte de dialogue avec **Fonte de l'application** sélectionnée vous est présentée dans l'illustration suivante.



Le G utilise les trois catégories de fontes différentes pour des parties spécifiques de son interface. Les fontes sont prédéfinies en fonction de la plate-forme de la manière suivante :

Utiliser la fonte par défaut : utilise la fonte par défaut, définie par le logiciel conformément aux différentes plates-formes comme il a été précédemment décrit.

Fonte personnalisée : cette case est automatiquement cochée si vous modifiez une des caractéristiques de fonte par le biais de l'option **Style de fonte...**

Style de fonte... : affiche une boîte de dialogue permettant de modifier les caractéristiques de fonte.



Remarque

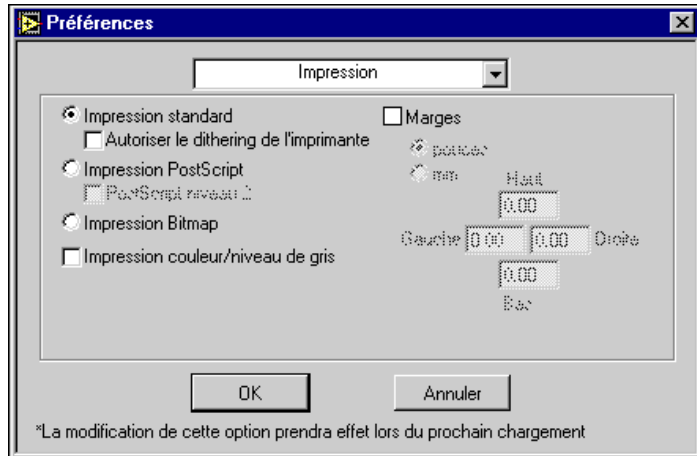
Les modifications apportées aux options de la boîte de dialogue de Préférences Fonte ne prennent effet que lors du redémarrage du logiciel.

Pour obtenir des informations spécifiques sur la manière dont les fontes sont prédéfinies ou pour obtenir des informations concernant la portabilité

des fontes, consultez la section *Différences de résolution et de fonte* du chapitre 29, *Portabilité et localisation*.

Préférences d'impression

La boîte de dialogue de Préférences Impression vous est présentée dans l'illustration suivante. Gardez à l'esprit que certaines options ne sont disponibles que sur certaines plates-formes.



(Windows, Macintosh) Impression standard : formate les données d'impression du VI (face-avant, diagramme, icône, etc.) et les imprime en utilisant les commandes de dessin standard. Utilisez cette option si vous souhaitez imprimer vers un fichier (ce qui suppose que votre driver d'imprimante supporte l'impression vers un fichier), si votre imprimante ne supporte pas l'impression PostScript, ou si vous voulez que votre driver d'imprimante effectue la conversion PostScript à la place du logiciel.

(Windows uniquement) Autoriser le dithering de l'imprimante : utilise le dithering de l'imprimante lors de la création des commandes de dessin. Applicable à l'impression en noir et blanc, le dithering de l'imprimante est une méthode qui consiste à ajouter des niveaux de gris à une image au lieu de simples zones en noir et blanc. Cet élément de menu ne dépend pas du type d'imprimante.

Impression PostScript : convertit les données d'impression du VI au format PostScript (.ps) et les envoie vers l'imprimante sous forme de données PostScript. Si vous possédez une imprimante PostScript et un driver d'imprimante PostScript, l'impression sera une image graphique.

Ne sélectionnez pas cet élément de menu si votre imprimante ou driver d'imprimante ne supporte pas l'impression PostScript.

PostScript Niveau 2 : signale au logiciel que l'imprimante connectée supporte le **PostScript Niveau 2**. Si cette case est cochée, le logiciel envoie le code **PostScript Niveau 2** à l'imprimante. Ne sélectionnez cette option que si votre imprimante supporte le **PostScript Niveau 2**.

(Windows, UNIX) Impression bitmap : crée un bitmap, dessine toutes les données de cette page dans un fichier bitmap et envoie le fichier bitmap à l'imprimante. Cette méthode rallonge le temps d'impression par rapport aux deux autres, mais produit une représentation du texte et des fontes plus précise que l'impression standard. Par contre, la résolution de l'impression n'atteint pas celle de l'impression PostScript. L'image bitmap est en couleur pour les imprimantes couleur et en noir et blanc pour les autres.

(Windows uniquement) Impression couleur/niveau de gris : cet élément de menu génère une sortie couleur/niveaux de gris de 8 bits/pixel (bpp), et remplace les paramètres de codage obtenus par le driver de l'imprimante. La plupart des imprimantes présentent correctement les couleurs, mais certains drivers d'imprimante laser présentent un codage de bits de 1 alors que l'imprimante peut en réalité gérer 8 bpp.

Marges : vous pouvez définir des marges absolues pour les impressions en millimètres ou pouces. Les quatre marges (haut, bas, gauche et droite) peuvent être spécifiées de manière séparée. Les marges sont limitées par les paramètres physiques de l'imprimante. Si vous définissez des marges de plus petites dimensions que celles acceptées par l'imprimante, les véritables marges seront le minimum accepté par l'imprimante. Les paramètres des marges de la boîte de dialogue de préférence affectent les nouveaux VIs créés ou convertis. Si vous souhaitez définir des marges pour un VI spécifique, utilisez **Configuration du VI**.



Remarque

Les modifications apportées aux options de la boîte de dialogue de Préférences Impression prennent effet immédiatement.

Préférences de l'historique

Comme il est expliqué dans le chapitre 27, *Gestion de vos applications*, chaque VI possède une fenêtre Historique qui affiche l'historique du développement du VI. Vous pouvez choisir les paramètres par défaut de la fenêtre Historique des nouveaux VIs à l'aide de la boîte de dialogue Préférences de l'historique.

La boîte de dialogue de Préférences Historique vous est présentée dans l'illustration suivante.



Les options de la boîte de dialogue Historique sont divisées en deux groupes. Le premier groupe contient cinq options qui indiquent le moment et la manière dont les entrées sont ajoutées à la fenêtre Historique des nouveaux VIs.

Le deuxième groupe d'options concerne la manière dont vous vous connectez au logiciel LabVIEW ou BridgeVIEW, ce qui à son tour détermine le nom inséré dans les en-têtes des commentaires entrés dans la fenêtre Historique du VI.

Les options qui concernent les entrées de la fenêtre Historique sont spécifiées pour *chaque* VI grâce à la boîte de dialogue **Configuration du VI»Documentation**. Pour plus d'informations sur l'accès aux options, consultez la section *Options de documentation* du chapitre 6, *Configuration des VIs et sous-VIs*.

Les options de cette boîte de dialogue sont les suivantes :

Ajouter une entrée à chaque enregistrement du VI : ajoute une entrée à l'historique du VI à chaque fois que vous l'enregistrez. Si vous ne saisissez pas de commentaire dans la zone **Commentaire** de la fenêtre Historique, seul un en-tête y est ajouté. (L'en-tête contient le numéro de révision si l'élément de menu plus bas dans cette boîte de dialogue est coché, ainsi que la date, l'heure et le nom du VI).

Proposer de saisir un commentaire lors de l'enregistrement du VI : affiche la fenêtre Historique pour la saisie de commentaires sur les modifications apportées depuis le précédent enregistrement. Cette option est utile si vous préférez commenter vos modifications à la fin de leur création plutôt que lors de l'édition. Si vous ne sélectionnez pas cette option, vous ne pourrez pas modifier l'historique du VI une fois que vous aurez sélectionné **Enregistrer** et ce jusqu'à la fin de l'enregistrement.

Proposer de saisir un commentaire lors de la fermeture du VI : identique à la précédente option, excepté le fait que le G ne vous invite à saisir un commentaire qu'à la fermeture du VI plutôt qu'à chaque enregistrement. Vous y êtes invité si le VI a été modifié depuis son dernier enregistrement, même si vous avez enregistré les modifications.



Remarque

Vous n'êtes pas invité à saisir de commentaires lorsque vous enregistrez ou fermez un VI si seules des modifications ont été apportées à l'historique.

Enregistrer les commentaires générés par l'éditeur : l'éditeur insère des commentaires dans la fenêtre Historique lorsque certains événements ont lieu. Les événements qui produisent des commentaires automatiques sont les conversions vers une nouvelle version de LabVIEW ou de BridgeVIEW, les modifications du sous-VI et celles du nom ou du chemin du VI.

Visualiser le numéro de révision dans la barre de titres : ajoute un numéro de révision à l'en-tête de la fenêtre Historique. Ce numéro commence par zéro et augmente à chaque fois que vous enregistrez le VI. Toutefois, il n'augmente pas si seules des modifications ont été apportées à l'historique.

Les options de login sont les suivantes :

(Windows et Macintosh uniquement) Login automatique avec le nom d'inscription : utilise le nom de l'utilisateur inscrit de l'application.

Confirmer le login lors du démarrage : vous invite à saisir le nom de l'utilisateur au démarrage de votre application. Le nom utilisateur peut être modifié à tout moment en sélectionnant **Edition»Nom utilisateur**. Avec cette option, l'invite affiche un menu de tous les noms utilisateurs précédemment entrés dans LabVIEW.

 **Remarque**

*Cette fonction ne concerne que LabVIEW. BridgeVIEW possède son propre système de sécurité de login et l'option **Edition»Nom utilisateur (Edit»User Name)** n'existe pas.*



Vous pouvez accéder à un autre nom utilisateur sous Windows ou sous Macintosh (le nom utilisateur système ou votre nom d'inscription à l'application), ou le nom utilisateur système sous UNIX sans avoir à le saisir en utilisant le menu déroulant présenté dans la précédente illustration. Vous pouvez également saisir un des noms de la liste.

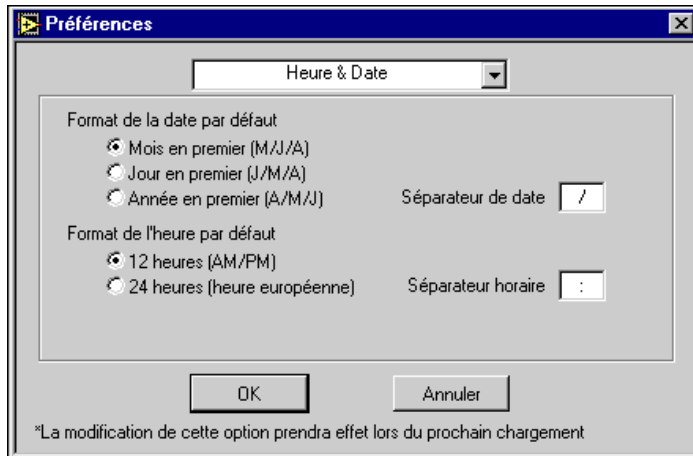
(Windows NT, Macintosh avec partage de fichiers uniquement, et UNIX) Login automatique avec le nom de l'utilisateur du système : en supposant qu'un nom utilisateur soit défini, le logiciel utilise le nom de login système de l'utilisateur en cours.

 **Remarque**

*Les modifications apportées aux options de la boîte de dialogue de **Préférences Historique** prennent effet immédiatement.*

Préférences pour l'affichage de la date et de l'heure

La boîte de dialogue de Préférences Heure & Date vous est présentée dans l'illustration suivante.



Les options de cette boîte de dialogue contrôlent les affichages par défaut de la date et de l'heure pour les affichages numériques des nouvelles commandes et des nouveaux indicateurs. Remplacez les valeurs par défaut dans cette boîte de dialogue pour un affichage numérique individuel, comme il est expliqué dans la section *Format et précision des affichages numériques* du chapitre 9, *Commandes et indicateurs numériques*.

Les options de cette boîte de dialogue sont les suivantes :

Format de la date par défaut : détermine si le mois, le jour ou l'année doit apparaître en premier dans les affichages numériques.

Format de l'heure par défaut : détermine si l'affichage de l'heure est basé sur une horloge de 12 heures ou de 24 heures.

Séparateur de date : détermine le caractère utilisé pour séparer le mois, le jour et l'année (dans l'ordre sélectionné dans l'option Format de la date par défaut).

Séparateur horaire : détermine le caractère utilisé pour séparer les heures des minutes.

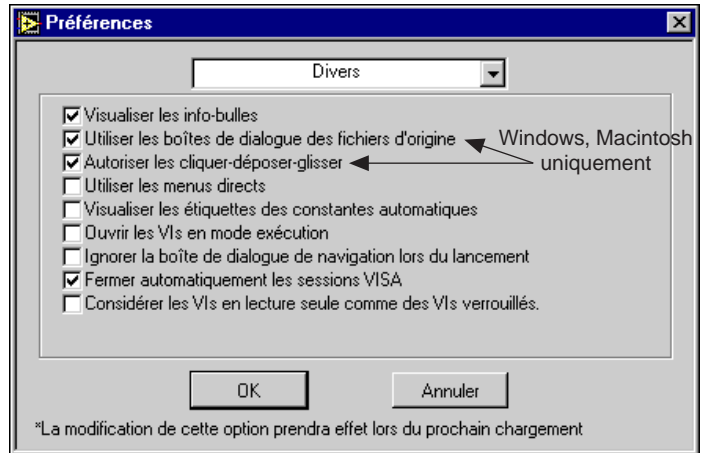


Remarque

Les modifications apportées aux options de la boîte de dialogue de Préférences Heure et Date prennent effet immédiatement.

Préférences diverses

La boîte de dialogue de Préférences Divers vous est présentée dans l'illustration suivante.



Les options de cette boîte de dialogue sont les suivantes :

Visualiser les info-bulles : active ou désactive l'affichage des info-bulles.

(Windows, Macintosh) Utiliser les boîtes de dialogue du fichier d'origine : utilise les boîtes de dialogue du fichier d'origine du système d'exploitation de manière à ce qu'elles ressemblent à celles des autres applications de votre ordinateur (à l'exception de quelques cas traités dans le paragraphe suivant). Lorsque cette préférence n'est pas sélectionnée, le logiciel utilise ses propres fichiers de boîtes de dialogue qui sont indépendantes de la plate-forme. Ces derniers comportent quelques fonctions pratiques telles que la liste des chemins récents et un nombre d'étapes réduit pour enregistrer les VIs dans les bibliothèques de VIs.

(Windows, Macintosh) Autoriser les cliquer-déposer-glisser : vous pouvez configurer le logiciel de manière à ce qu'un seul clic de la souris active et sélectionne un objet dans une fenêtre inactive. Cette opération réduit de moitié le nombre de clics de souris nécessaires (normalement, le premier clic active la fenêtre et le deuxième clic est considéré comme un clic dans la fenêtre).

Utiliser les menus directs : avec cette option, vous pouvez naviguer dans les menus en utilisant la souris sans maintenir le bouton enfoncé.

Visualiser les étiquettes des constantes automatiques : cette option active ou désactive la création des étiquettes des constantes créées automatiquement. Lorsque vous l'activez, les étiquettes affichent automatiquement le nom du terminal à partir duquel la constante a été créée.

Ouvrir les VIs en mode Exécution : ouvre les VIs en mode Exécution plutôt qu'en mode Edition.

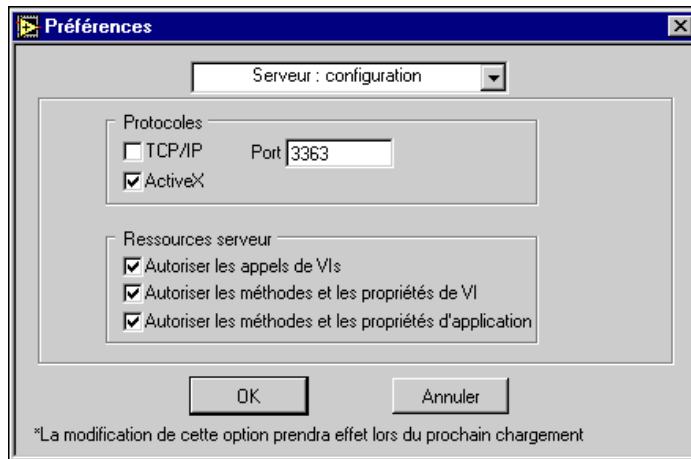
Ignorer la boîte de dialogue de navigation lors du lancement : ignore la fenêtre initiale qui permet la navigation vers les boîtes de dialogue communes et ouvre directement le VI Sans titre.

 Remarque

Les modifications apportées aux options de la boîte de dialogue de Préférences Divers prennent effet immédiatement.

Serveur : configuration

La boîte de dialogue Serveur : configuration vous est présentée dans l'illustration suivante.



Ces options spécifient par quels protocoles de communication les autres applications accèdent au VI Serveur : TCP/IP ou ActiveX. Si vous activez **TCP/IP**, vous devez fournir le numéro du **Port** que les applications clientes doivent utiliser pour se connecter au serveur. Lorsque vous donnez la permission aux applications de se connecter en utilisant TCP/IP, il est préférable de préciser quels hôtes Internet ont accès au serveur. Pour plus de détails, consultez la section *Serveur : accès TCP/IP* de ce chapitre. Pour

plus d'informations sur l'interface ActiveX du VI Serveur, consultez le chapitre 21, *VI Serveur*.

Dans la boîte de dialogue Serveur : configuration, vous pouvez également spécifier les ressources du serveur, parmi celles présentées ci-dessous, disponibles aux applications clientes.

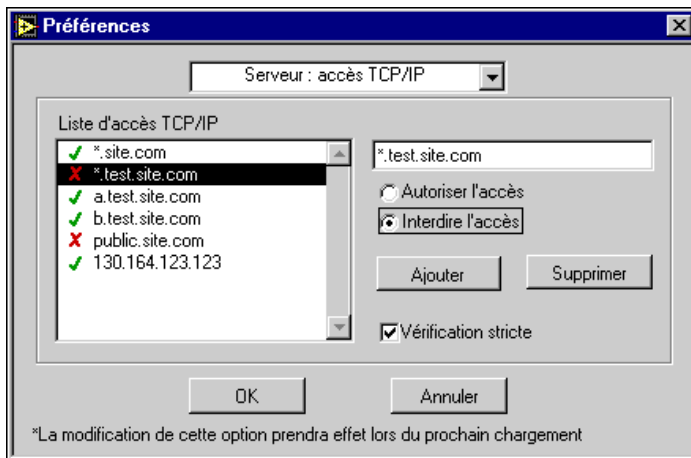
- **Autoriser les appels du VI** donne la possibilité aux applications d'effectuer des appels aux VIs exportés sur le serveur. Dans ce cas, il est préférable de préciser quels VIs sont exportés en utilisant la page Préférences du VI exporté. Pour plus d'informations, consultez la section *Serveur : VIs exportés*, dans ce chapitre.
- **Autoriser les méthodes et les propriétés du VI** donne la possibilité aux applications de lire et de définir les propriétés des VIs sur le serveur. Encore une fois, vous pouvez préciser les VIs en utilisant la page Préférences du VI exporté. Pour plus d'informations, consultez la section *Serveur : VIs exportés*, dans ce chapitre.
- **Autoriser les méthodes et les propriétés de l'application** donne la possibilité aux applications de lire et de définir les propriétés du serveur.

Les paramètres par défaut du serveur sont **ActiveX** activé et **TCP/IP** désactivé. Par défaut, l'option **Appels du VI** est activés, mais **Méthodes et propriétés du VI** et **Méthodes et propriétés de l'application** sont désactivées.

Serveur : accès TCP/IP

Lorsque vous autorisez aux applications clientes d'accéder au VI Serveur grâce au protocole TCP/IP, il est préférable de préciser quels hôtes Internet ont accès au serveur.

La boîte de dialogue **Serveur : accès TCP/IP** vous est présentée dans l'illustration suivante.



En utilisant **Serveur : accès TCP/IP**, vous pouvez spécifier quels clients ont accès au VI Serveur. La Liste d'accès TCP/IP décrit les clients pour lesquels l'accès au serveur est autorisé ou refusé. Pour modifier une entrée, sélectionnez-la dans la liste et entrez les modifications dans la zone de texte située à droite. Cliquez sur le bouton radio **Autoriser l'accès** afin de donner la possibilité au client d'accéder au serveur. Cliquez sur le bouton radio **Refuser l'accès** dans le cas contraire. Cliquez sur le bouton **Ajouter** pour insérer une nouvelle entrée à la suite de la sélection en cours. Cliquez sur le bouton **Supprimer** pour supprimer l'entrée sélectionnée de la liste. Cliquez et faites glisser une entrée pour modifier sa position dans la Liste d'accès TCP/IP. Lorsque l'accès est autorisé, une encoche s'affiche à côté de l'entrée. Dans le cas contraire, un **X** remplace l'encoche. Si aucun symbole n'apparaît, la syntaxe de l'entrée est incorrecte.

Lorsqu'un client tente d'ouvrir une connexion au serveur, le serveur examine les entrées de la Liste d'accès TCP/IP pour déterminer si son accès est autorisé. Si une entrée de la liste correspond à l'adresse du client, le serveur autorise ou refuse l'accès en fonction du choix que vous avez établi. Si une entrée suivante correspond également à l'adresse du client, cette autorisation est utilisée à la place de la précédente. Par exemple, dans

l'illustration précédente, `a.test.site.com` et `b.test.site.com` ont une autorisation d'accès, même si la liste précise (par le caractère générique `*`) que toutes les adresses qui se terminent par `.test.site.com` doivent être refusées. Si aucune entrée ne correspond à l'adresse du client, l'accès est alors refusé. Pour plus d'informations sur le caractère générique `*` et l'autorisation d'accès correspondant aux entrées, reportez-vous au tableau 7-1.

Une adresse Internet (IP), telle que `130.164.123.123`, peut avoir un ou plusieurs noms de domaine associés, du genre `www.natinst.com`. La conversion d'un nom de domaine en adresse IP correspondante est appelée *résolution de nom*. La conversion d'une adresse IP en nom de domaine est appelée *recherche de nom*.

Les recherches et les résolutions de nom sont effectuées grâce aux appels système qui accèdent au Serveur de nom de domaine (DNS) sur Internet. Une recherche et une résolution de nom peuvent échouer lorsque le système n'a pas accès à un serveur DNS, ou lorsque l'adresse ou le nom est incorrect. Un problème de résolution survient lorsqu'une entrée contient un nom de domaine qui ne peut pas être résolu en une adresse IP. Un problème de recherche survient lorsqu'une entrée contient un nom de domaine incomplet, tel que `*.natinst.com`, et que la recherche de l'adresse IP du client échoue.

L'option **Vérification stricte** détermine la manière dont le serveur traite les entrées de la liste d'accès qui ne correspondent pas à l'adresse IP d'un client à cause d'un problème de résolution ou de recherche. Lorsque cette option est activée, une entrée de la liste d'accès refusé dans la Liste d'accès TCP/IP qui rencontre un problème de résolution est traitée comme si elle correspondait à l'adresse IP du client. Lorsque **Vérification stricte** est désactivé, une entrée de la liste d'accès qui rencontre un problème de résolution est ignorée.

Pour spécifier une adresse d'hôte Internet, entrez son nom de domaine ou son adresse IP. Utilisez le caractère générique `*` lorsque vous spécifiez plusieurs adresses d'hôtes Internet. Par exemple, vous pouvez spécifier tous les hôtes du domaine `domain.com` avec l'entrée `*.domain.com`. Vous pouvez spécifiez tous les hôtes du sous-réseau dont les deux premiers octets sont `130.164` avec l'entrée `130.164.*`. L'entrée `*` correspond à toutes les adresses.

Le tableau suivant vous propose des exemples d'entrées de la Liste d'accès TCP/IP.

Tableau 7-1. Serveur : accès TCP/IP

Chaînes d'accès	Correspondances
*	Tous les hôtes
test.site.com	L'hôte dont le nom de domaine est test.site.com
*.site.com	Tous les hôtes dont le nom de domaine se termine par .site.com
130.164.123.123	L'hôte dont l'adresse IP est 130.164.123.123
130.164.123.*	Tous les hôtes dont l'adresse IP commence par 130.164.123.

Dans l'illustration précédente, tous les hôtes du domaine site.com ont accès au serveur, à l'exception de tous les hôtes du domaine test.site.com. De plus, les hôtes a.test.site.com, b.test.site.com et 130.164.123.123 ont également accès au serveur. L'hôte public.site.com n'a pas d'accès bien qu'il soit dans le domaine site.com.

Les paramètres d'accès TCP/IP par défaut n'autorisent l'accès qu'aux clients de la machine du serveur.

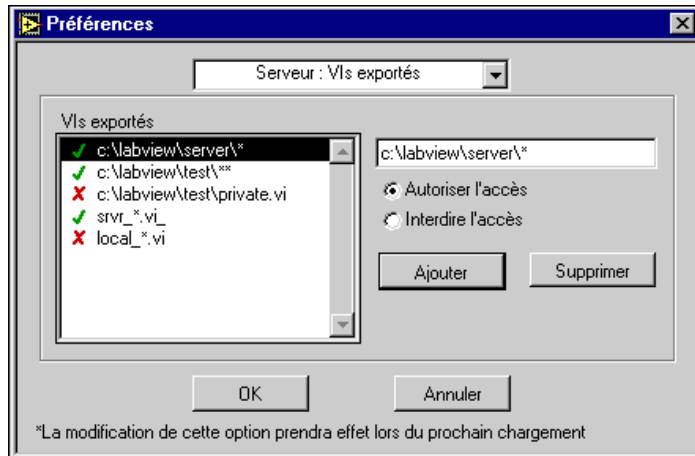
 **Remarque**

Si le VI Serveur est exécuté sur un système qui n'a pas d'accès à un serveur DNS, n'utilisez pas les entrées du nom de domaine de la Liste d'accès TCP/IP. Les requêtes de résolution du nom de domaine ou de l'adresse IP échoueront, ce qui ralentira le système. Pour améliorer les performances, placez les entrées les plus fréquemment rencontrées dans le bas de la Liste d'accès TCP/IP.

Serveur : VIs exportés

Si vous autorisez aux applications clientes l'accès aux VIs du VI Serveur, il est préférable de spécifier à quels VIs ces applications peuvent accéder.

La boîte de dialogue **Serveur : VIs exportés** vous est présentée dans l'illustration suivante.



Les éléments de menu **Serveur : VIs exportés** vous permettent de spécifier à quels VIs les autres applications peuvent accéder par le biais du VI Serveur. Pour modifier une entrée de la liste des VIs exportés, sélectionnez-la dans la liste et entrez les modifications dans la zone de texte située à droite. Pour spécifier si des clients ont ou n'ont pas accès à ce VI, cliquez sur les boutons radio **Autoriser l'accès** ou **Refuser l'accès**. Cliquez sur le bouton **Ajouter** pour insérer une nouvelle entrée à la suite de l'entrée sélectionnée. Cliquez sur le bouton **Supprimer** pour supprimer l'élément sélectionné de la liste. Cliquez et faites glisser une entrée pour modifier sa position dans la liste des VIs exportés. Lorsque l'accès est autorisé, une encoche s'affiche à côté de l'entrée. Dans le cas contraire, un **X** remplace l'encoche. Si aucun symbole n'apparaît, la syntaxe de l'entrée est incorrecte.

Chaque entrée de la liste décrit un nom ou un chemin de VI et peut contenir des caractères génériques. Les entrées qui contiennent des séparateurs de chemin sont comparées aux chemins du VI, alors que les autres sont comparées aux noms du VI uniquement. Lorsqu'un client tente d'accéder à un VI, le serveur examine la liste des VIs exportés pour déterminer si son accès est autorisé. Si une entrée de la liste correspond au VI demandé, le serveur autorise ou refuse l'accès en fonction du choix que vous avez établi. Si une entrée suivante correspond également au VI, cette autorisation est utilisée à la place de la précédente. Si aucun VI de la liste ne correspond au VI demandé, l'accès est refusé.

Vous pouvez utiliser des caractères génériques dans la liste des VIs exportés afin qu'une entrée de la liste corresponde à plusieurs VIs. Les caractères génériques sont les suivants :

'?'	Correspond exactement à un caractère arbitraire, excepté pour le séparateur de chemin
'*'	Correspond à zéro ou plusieurs caractères arbitraires, excepté pour le séparateur de chemin
'**'	Correspond à zéro ou plusieurs caractères arbitraires, y compris le séparateur de chemin

Si vous voulez faire correspondre un VI à un nom qui contient un caractère générique, vous devez éviter ce caractère en utilisant '\ ' (**Macintosh** et **UNIX**) ou '\\ ' (**Windows**).

Le tableau suivant propose des exemples d'entrées de liste de VI exporté. Les exemples utilisent les séparateurs de chemin de l'UNIX.

Tableau 7-2. Serveur : Entrées de liste de VI exporté

Chaîne d'accès au VI	Correspondance
*	Tous les VIs
/usr/labview/*	Tous les VIs du répertoire /usr/labview/
/usr/labview/**	Tous les VIs du répertoire /usr/labview/ et de tous ses sous-répertoires
test.vi	Tout VI nommé Test.vi.
OK\?	Tout VI comportant le nom OK?

Dans l'illustration précédente, tous les VIs du répertoire c:\labview\server sont exportés. Tous les VIs du répertoire c:\labview\test et de tous ses sous-répertoires sont également exportés, à l'exception du VI c:\labview\test\private.vi. De plus, tout VI qui commence par la chaîne srvr_ et se termine par .vi est exporté. Aucun VI qui commence par la chaîne local_ et se termine par .vi n'est exporté, même s'il se trouve dans le répertoire c:\labview\server.

Les paramètres par défaut des VIs exportés autorisent l'accès à tous les VIs.

Comment sont stockées les préférences ?

En général, vous n'avez pas à éditer les informations des préférences manuellement ni à connaître leur format exact car la boîte de dialogue Préférences prend tout en charge. Les préférences sont stockées de manière différente sur chaque plate-forme. C'est ce que nous allons vous décrire dans les paragraphes suivants.

(Windows) Si vous êtes utilisateur de LabVIEW, vous trouverez les informations sur les préférences dans le fichier `LabVIEW.INI` du répertoire `LabVIEW`. Le format du fichier est identique aux autres fichiers `.INI`, tels que `WIN.INI`. Il commence par un marqueur de section `[LabVIEW]`.

Il est suivi par des étiquettes de variables et leurs valeurs, telles que `offscreenUpdates=True`. Si la valeur d'une configuration n'est pas définie dans `LABVIEW.INI`, LabVIEW recherche une section intitulée `[LabVIEW]` dans le fichier `WIN.INI` et, s'il la trouve, il contrôle la valeur de la configuration de ce fichier.

Si vous êtes un utilisateur de BridgeVIEW, vous trouverez les informations sur les préférences dans le fichier `BridgeVIEW.INI` du répertoire `BridgeVIEW`. Le format de ce fichier est identique aux autres fichiers `.INI`. Il commence par un marqueur de section `[BridgeVIEW]`.



Remarque

Les références aux fichiers `WIN.INI` ne s'appliquent qu'à Windows 3.1. Ces informations ne vous concernent pas si vous êtes utilisateur de BridgeVIEW.

Vous pouvez également spécifier un fichier de préférences sur la ligne de commande lors du lancement de LabVIEW ou de BridgeVIEW. Par exemple, pour utiliser un fichier nommé `lvrc` au lieu de `labview.ini`, tapez `labview -pref lvrc`.

(Macintosh) Par défaut, LabVIEW crée un fichier texte dans le dossier Préférences du Dossier système nommé `LabVIEW Preferences`.

Si vous le souhaitez, copiez le fichier Préférences de LabVIEW dans le dossier LabVIEW. Au démarrage, LabVIEW recherche toujours un fichier Préférences dans le dossier LabVIEW. S'il ne le trouve pas, il le recherche ensuite dans le dossier Préférences, et s'il reste introuvable, il en crée un nouveau dans ce même dossier. En déplaçant les fichiers Préférences du bureau vers le dossier LabVIEW, vous pouvez créer plusieurs Préférences pour différents utilisateurs.

(UNIX) Les informations sur les préférences sont normalement stockées dans un fichier texte de votre répertoire nommé `.labviewrc`. Si vous modifiez un paramètre dans la boîte de dialogue Préférences, le G enregistre les informations dans ce fichier. Ce qui suit ne concerne que les utilisateurs qui veulent en savoir plus sur le format et les règles de recherche des informations de préférences.

Les entrées des préférences sont composées du *nom de la préférence* suivi par un deux-point et une valeur. Le nom de la préférence correspond au nom exécutable suivi par un point (.) et un symbole. Lorsque LabVIEW recherche des noms de préférences, il ignore la casse. Vous pouvez encadrer la valeur de la préférence par de simples ou doubles guillemets. Par exemple, pour utiliser par défaut une double précision et utiliser un chemin qui recherche de manière récursive un répertoire particulier, spécifiez les entrées de préférences suivantes.

```
labview.defPrecision : double
labview.viSearchPath : "/usr/lib/labview/*"
```

LabVIEW recherche également des préférences dans un fichier nommé `labviewrc` dans le répertoire des applications. Par exemple, si vous avez installé vos fichiers LabVIEW dans `/opt/labview`, les préférences sont recherchées à la fois dans les fichiers `/opt/labview/labviewrc` et `.labviewrc` de votre répertoire principal.

Les entrées du fichier `.labviewrc` remplacent les entrées conflictuelles du fichier `labviewrc`. Vous pouvez utiliser ce fichier de préférences globales pour stocker des valeurs communes à tous les utilisateurs. Ce qui peut être, par exemple, le cas du chemin de recherche de VI.

Vous pouvez également spécifier un fichier de préférences dans la ligne de commande lors du démarrage de LabVIEW. Par exemple, pour utiliser un fichier nommé `lvrc`, tapez `labview -pref lvrc`.

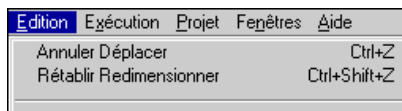


Avertissement

Le fichier `.labviewrc` est lu, même dans ce cas. Gardez à l'esprit que, dans cet exemple, les modifications apportées dans la boîte de dialogue Préférences sont enregistrées dans le fichier `lvrc`.

Annuler

En utilisant les fonctions **Edition»Annuler** et **Edition»Revenir**, vous pouvez annuler une action immédiatement après l'avoir effectuée et pouvez la rétablir une fois celle-ci annulée. Le raccourci clavier de la fonction **Annuler** est <Ctrl-Z> (ou <Cmd-Z> sur Macintosh) et celui de la fonction **Revenir** est <Ctrl-Maj-Z> (ou <Cmd-Maj-Z> sur Macintosh). Aux éléments du menu **Edition**, que sont **Annuler** et **Revenir**, est ajoutée une brève description de l'action (un ou deux mots) qui peut être annulée ou rétablie. Dans l'illustration suivante, si vous sélectionnez **Annuler redimensionnement**, l'élément revient à sa taille d'origine.



Définissez le nombre maximum d'actions que vous pouvez annuler ou rétablir dans **Préférences»Diagramme»Nombre maximum d'annulations par VI**. Si le nombre d'actions est zéro, les fonctions **Annuler** et **Revenir** sont désactivées. Pour plus d'informations sur le paramétrage du nombre maximum d'étapes d'annulation par VI, consultez la section *Préférences pour le diagramme*, de ce chapitre.

Chaque VI enregistre ses propres annulations. Par conséquent, le nombre que vous tapez concerne chaque VI. Définissez un petit nombre si vous souhaitez économiser les ressources mémoire.

Les appelants du VI se mettent automatiquement à jour lorsque le VI change son interface. Dans cette situation, toutes les informations d'annulation des appelants disparaissent. Cela se produit lorsque vous effectuez une des opérations suivantes.

- Modification d'un modèle de cadre connecteur de VI ou modification du type de données d'une commande sur le cadre connecteur.
- Modification de l'apparence d'une définition de type (uniquement si elle est stricte), modification du type ou de la précision de données, ou modification d'une définition de type en commande personnalisée.
- Modification du type de données ou du nom d'une commande dans un VI global, ou ajout ou suppression d'une commande d'un VI global.

Personnalisation des palettes Commandes et Fonctions

Le G possède un ensemble d'outils puissants pour la personnalisation des palettes **Commandes** et **Fonctions**. Pour les utiliser, effectuez une des opérations suivantes :

- Ajoutez vos propres VIs et commandes aux palettes en les enregistrant dans les répertoires `user.lib` ou `instr.lib`.
- Ajoutez vos propres VIs et commandes aux palettes en utilisant l'option **Edition»Editer les palettes...**
- Configurez les palettes de manière à ce qu'elle soient automatiquement modifiées lorsque vous ajoutez ou supprimez des fichiers de certains répertoires spécifiques.
- Réorganisez les palettes intégrées afin que les fonctions fréquemment utilisées soient plus accessibles.
- Configurez différents affichages pour les différents utilisateurs. Masquez certaines fonctions pour alléger l'utilisation du G pour un utilisateur et mettez toutes les capacités du programme à la disposition d'un autre utilisateur.
- Sous Windows et Macintosh, la **palette Fonctions est masquée lorsqu'une face-avant constitue la fenêtre active et la palette Commandes est masquée lorsqu'un diagramme constitue la fenêtre active. Si vous préférez afficher toutes les palettes en permanence, éditez votre fichier de préférences en ajoutant une ligne définissant la préférence `showAllPalettes` en `True`.**

Ajout de VIs et de commandes aux fichiers `user.lib` et `instr.lib`

La méthode la plus simple pour ajouter de nouvelles entrées aux palettes **Commandes** et **Fonctions** consiste à les enregistrer dans le répertoire `user.lib`. Lorsque vous redémarrez le G, la sous-palette **Bibliothèque utilisateur** de la palette **Fonctions** contient alors les sous-palettes de chaque répertoire, un fichier `.llb`, ou `.mnu` dans `user.lib` et les entrées pour chaque fichier dans `user.lib`.

La palette **Drivers d'instruments** de la palette **Fonctions** correspond au répertoire `instr.lib`. Vous pouvez mettre, si vous le souhaitez, des drivers d'instruments dans ce répertoire pour les rendre plus facilement accessibles à partir des palettes.

**Remarque**

Le répertoire `vi.lib` contient des fichiers de National Instruments qui sont écrasés lors de la mise à jour du logiciel. N'enregistrez donc pas vos propres fichiers dans `vi.lib`.

Installation et modification des affichages de palettes

Les informations concernant les palettes **Commandes** et **Fonctions** sont stockées dans le répertoire menus du répertoire de votre application. Le répertoire menus contient des répertoires qui correspondent à chaque affichage créé ou installé. Si vous êtes sur un réseau, vous voudrez peut-être définir des répertoires menus individuels pour chaque utilisateur. Vous pouvez ajouter une ligne à votre fichier de préférences qui définit la préférence `menusDir` sur un autre chemin, ce dernier étant unique pour chaque fichier de préférences de chaque utilisateur.

Cette organisation facilite le transfert des affichages de palettes à d'autres utilisateurs. Pour remettre une copie de l'affichage à quelqu'un d'autre, donnez-lui une copie du répertoire de l'affichage du répertoire menus. Une fois qu'il l'a placée dans son répertoire menus, il l'aura à sa disposition au prochain démarrage.

Pour basculer vers un autre affichage de palette, sélectionnez **Edition» Sélectionner le type de palette...** Sélectionnez ensuite l'affichage que vous voulez dans le menu qui apparaît.

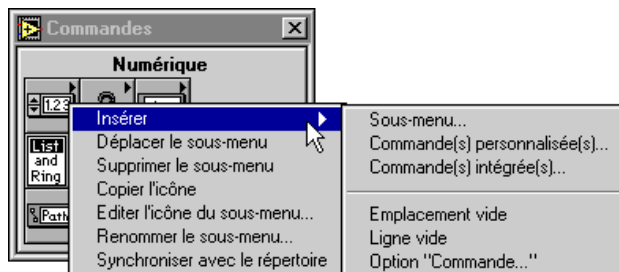
Pour créer votre propre affichage de palette, reportez-vous à la section suivante *Editeur de palettes*. Pour plus de détails sur la manière dont les affichages sont organisés, consultez la section *Fonctionnement des affichages des palettes* de ce chapitre.

Editeur de palettes

Pour mieux contrôler la disposition et le contenu des palettes **Commandes** et **Fonctions**, utilisez l'option **Edition»Editer les palettes**. Lorsque vous sélectionnez cette option, vous entrez dans l'Editeur de palettes.

Dans cet éditeur, vous pouvez réorganiser le contenu des palettes en faisant glisser les objets vers de nouveaux emplacements. Si vous souhaitez supprimer, personnaliser, ou insérer des objets, ouvrez un menu local sur

une palette comme indiqué dans l'illustration suivante, ou ouvrez un menu local sur un objet à l'intérieur de la sous-palette.



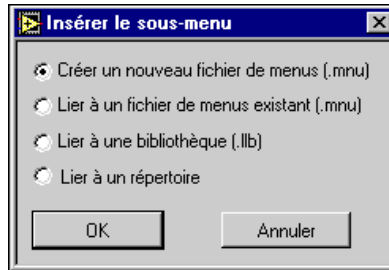
Les options du menu local vous permettent de modifier tout ce qui se trouve dans la palette **Bibliothèque utilisateur** (équivalent à `user.lib`) ou dans le menu **E/S d'instrument** (équivalent à `instr.lib`). Si vous souhaitez éditer les palettes **Commandes** ou **Fonctions** de haut niveau ou n'importe quel autre menu prédéfini, vous devez d'abord créer un nouvel affichage en sélectionnant **nouvelle configuration...** du menu déroulant **type de palette** de la boîte de dialogue d'édition des palettes. Une fois que vous avez sélectionné une nouvelle configuration, tout menu que vous modifiez et qui fait partie du jeu de menus par défaut sera copié dans le répertoire de votre affichage du répertoire `menus` avant d'effectuer les modifications. Cette protection des palettes intégrées vous assure la possibilité d'expérimenter de nouvelles palettes sans corrompre l'affichage par défaut.

Si vous souhaitez ajouter un nouvel objet dans une nouvelle ligne ou colonne d'une sous-palette, ouvrez un menu local à droite ou en bas de la sous-palette. Vous pouvez également créer de nouvelles lignes ou colonnes en faisant glisser des objets vers la droite ou vers le bas de la palette.

Création de sous-palettes

Lorsque vous ajoutez une palette, vous pouvez la déplacer vers un nouvel emplacement, éditer l'icône de la sous-palette, ou renommer la palette en utilisant l'Editeur de palettes.

Si vous souhaitez créer une palette à partir d'un modèle ou "accrocher" une palette qui ne se trouve pas dans `user.lib` ou `vi.lib`, vous pouvez utiliser la fonction **Insérer»Sous-menu...** du menu local de l'Editeur de palettes. Lorsque vous sélectionnez cette option, la boîte de dialogue suivante s'affiche.



Vous pouvez stocker des informations de sous-menu dans des Bibliothèques de VIs ou dans des fichiers .mnu. Un fichier de menu ou un fichier .LLB peut contenir une palette **Fonctions** et une palette **Commandes**.

Sélectionnez **Créer un nouveau fichier de menu** pour insérer une nouvelle palette vide. Vous êtes alors invité à saisir un nom de palette et un nom pour le fichier qui la contiendra. Il est recommandé d'ajouter une extension .mnu au fichier pour indiquer qu'il s'agit d'un menu (palette). De plus, il vous est recommandé de stocker le fichier .mnu dans le répertoire d'affichage du répertoire menus ou dans le même répertoire qui contient les commandes ou VIs que le menu représente le mieux.

Sélectionnez **Lier à un fichier de menu existant** si vous voulez ajouter le fichier de palette Commandes ou Fonctions à une palette existante.

Sélectionnez **Lier à un répertoire** si vous voulez créer une palette disposant d'entrées pour tous les fichiers du répertoire. Cette sélection crée également de manière récursive des sous-palettes pour chacun des sous-répertoires, bibliothèques VI, ou fichiers .mnu contenus dans le répertoire. Ces palettes se mettent automatiquement à jour si vous ajoutez de nouveaux fichiers ou si vous en supprimez dans les répertoires sélectionnés. Activez ou désactivez le paramètre de mise à jour automatique d'une sous-palette en sélectionnant l'option de menu local **Mise à jour automatique à partir du répertoire**. En fait, cet élément de menu crée des fichiers de menu (palette) à l'intérieur de chaque sous-répertoire.

Sélectionnez **Lier à une bibliothèque** si vous voulez lier la palette **Commandes** ou **Fonctions** à une bibliothèque de VI. De même que pour la précédente configuration, la palette d'une bibliothèque se met automatiquement à jour lorsque vous y ajoutez des fichiers.

Notez que vous pouvez librement mélanger les VIs, les fonctions, et les sous-palettes à l'intérieur d'une palette. En outre, une même palette peut contenir des VIs de différentes origines.

Déplacement de sous-palettes

Etant donné que le fait de cliquer sur une sous-palette l'ouvre, vous ne pouvez pas déplacer une sous-palette en la faisant glisser. Par conséquent, sélectionnez l'élément de menu **Déplacer le sous-menu** de son menu local. Le raccourci clavier de cette action consiste à maintenir appuyée la touche <Ma j> et à cliquer sur une sous-palette pour la déplacer au lieu de l'ouvrir.

Fonctionnement des affichages des palettes

Les fichiers `.mnu` et les fichiers de bibliothèque de VI sont des fichiers binaires qui peuvent contenir une palette **Commandes** et une palette **Fonctions**. Les deux types de fichiers contiennent également une icône pour chacune de ces palettes. Chaque sous-menu que vous créez doit être stocké dans un fichier séparé.

Lorsque vous sélectionnez un affichage, le G recherche un répertoire dans le répertoire `menus`. Il conçoit les palettes **Commandes** et **Fonctions** de haut niveau à partir du fichier `root.mnu` qui se trouve à l'intérieur de ce répertoire. Les palettes de haut niveau sont alors liées aux autres fichiers `.mnu` ou aux bibliothèques de VI associées à chaque sous-palette de la palette.

Si vous réalisez une liaison à un répertoire, le G contrôle si le répertoire contient un fichier `dir.mnu`. Si c'est le cas, le répertoire utilise ce fichier `.mnu` comme sous-palette. Sinon, votre application crée un fichier `dir.mnu` basé sur le contenu du répertoire. Pour chaque VI (ou commande), une nouvelle entrée est créée. Pour chaque sous-répertoire, fichier `.mnu`, ou bibliothèque nouvelle de VI, votre application crée une nouvelle sous-palette.

Le G met automatiquement à jour les palettes d'une bibliothèque de VI au fur et à mesure que vous ajoutez ou supprimez des fichiers. Vous pouvez, si vous le souhaitez, définir un fichier .mnv qui se met à jour en fonction du contenu d'un répertoire. Pour modifier cette configuration, sélectionnez **Synchroniser avec le répertoire** dans le menu local de l'icône de la sous-palette. Bien qu'il soit recommandé de garder le fichier .mnv dans le même répertoire auquel il est associé, vous pouvez en fait sélectionner n'importe quel répertoire. En plus de refléter un répertoire spécifique, vous pouvez mettre des VIs supplémentaires ou des commandes d'autres répertoires dans la même palette. Vous pouvez également supprimer des VIs de la palette en sélectionnant **Effacer l'élément** dans le menu local.

Objets de la face-avant

Cette section contient des informations sur les objets de la face-avant, les commandes et indicateurs et les commandes ActiveX.

La Partie II, *Objets de la face-avant*, contient les chapitres suivants :

- Le chapitre 8, *Introduction aux objets de la face-avant*, présente la face-avant et ses deux composantes : les commandes et les indicateurs. Il vous explique également comment importer des graphiques provenant d'autres applications afin de les utiliser dans vos commandes.
- Le chapitre 9, *Commandes et indicateurs numériques*, décrit comment créer, utiliser et configurer les divers styles de commandes et indicateurs numériques.
- Le chapitre 10, *Commandes et indicateurs booléens*, décrit comment créer, utiliser et configurer les commandes et indicateurs booléens.
- Le chapitre 11, *Commandes et indicateurs de chaîne de caractères*, décrit comment utiliser les commandes et indicateurs de chaîne de caractères, ainsi que les tables. Vous pouvez accéder à ces objets au moyen de la palette **Commandes»Chaîne de caractères & table**.
- Le chapitre 12, *Commandes et indicateurs de chemin et refnum*, décrit comment utiliser les commandes de chemin de fichier et les refnums, qui sont disponibles à partir de la palette **Commandes»Chemin & Refnum**.
- Le chapitre 13, *Commandes et indicateurs de type liste et menu déroulants*, décrit les commandes et indicateurs de type liste déroulante et de type menu déroulant, qui sont disponibles à partir de la palette **Commandes»Listes & menus déroulants**.
- Le chapitre 14, *Commandes et indicateurs de tableau et de cluster*, explique comment utiliser les tableaux et les clusters. Vous pouvez accéder aux tableaux et clusters à partir de la palette **Commandes»Tableaux et Clusters**.

- Le chapitre 15, *Indicateurs et commandes de graphe et de graphe déroulant*, explique comment créer et utiliser les indicateurs de graphes et de graphes déroulants de la palette **Commandes»Graphe**.
- Le chapitre 16, *Commandes ActiveX*, décrit les possibilités du Conteneur ActiveX, qui permettent des interactions entre des programmes écrits en G et les autres applications.

Introduction aux objets de la face-avant

Ce chapitre présente la face-avant et ses deux composantes que sont les commandes et les indicateurs. Il vous explique également comment importer des graphiques provenant d'autres applications afin de les utiliser dans vos commandes.

Construction de la face-avant

Les commandes et les indicateurs de la face-avant sont les terminaux interactifs d'entrée et de sortie du VI. Vous devez utiliser les commandes pour fournir des données à un VI. Les indicateurs, eux, affichent les données générées par le VI. Cette section présente quelques options d'édition communes à l'ensemble des indicateurs et des commandes.

La palette **Commandes** de la face-avant se présente de la manière suivante.



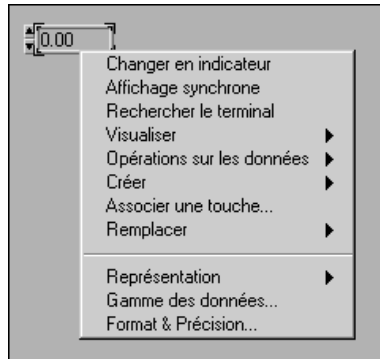
Si vous placez votre curseur sur un des carrés de la palette, le nom de l'ensemble de commandes s'affiche dans la partie supérieure de la fenêtre.

Les types de commandes de la palette, de gauche à droite en partant du haut, sont les suivants :

- **Commandes numériques** : pour la saisie et l’affichage des quantités numériques.
- **Commandes booléennes** : pour la saisie et l’affichage des valeurs True/False.
- **Commandes de chaîne de caractères & table** : pour la saisie et l’affichage de texte.
- **Commandes de type listes et menus déroulants** : pour l’affichage et/ou la sélection d’une liste d’options.
- **Commandes de type tableaux et clusters** : pour le regroupement d’ensemble de données.
- **Commandes de type graphe** : pour le tracé de données numériques sous forme de graphe.
- **Commandes de type chemin et Refnum** : pour la saisie et l’affichage des chemins de fichier et pour le transfert de refnums d’un VI à un autre.
- **Décorations** : pour l’ajout d’images afin de personnaliser la face-avant. Ces objets ne concernent que la décoration et n’affichent pas de données.
- **Commandes utilisateurs** : pour l’enregistrement automatique des commandes personnalisées dans le fichier `user.lib`.
- **ActiveX** : pour l’amélioration du support d’ActiveX. Ces objets comprennent le Container ActiveX et la Variable ActiveX.
- **Sélectionner une commande...** : pour choisir une commande personnalisée conçue par l’utilisateur.

Options des commandes et indicateurs de la face-avant

Lorsque vous ouvrez un menu local sur une commande ou un indicateur de la face-avant lors de l’édition d’un VI, vous obtenez un menu identique à celui présenté dans l’illustration suivante. Les options qui se situent au-dessus de la ligne du menu local sont communes à l’ensemble des commandes et des indicateurs. Les options situées au-dessous de la ligne comme, par exemple, **Représentation**, **Gamme des données...**, et **Format & Précision...** sont personnalisées en fonction de l’objet.



Les objets de la palette **Commandes** sont configurés au départ comme étant des commandes ou des indicateurs en fonction de la manière dont ils sont généralement utilisés. Par exemple, si vous choisissez un interrupteur à bascule de la palette **Booléen**, il s'affiche sur la face-avant comme une commande, l'interrupteur à bascule étant généralement utilisé pour réaliser une entrée. Réciproquement, si vous sélectionnez une LED, il s'affiche sur la face-avant comme un indicateur, la LED étant généralement utilisée pour réaliser une sortie. Toutefois, vous pouvez redéfinir toutes les commandes en indicateurs et tous les indicateurs en commandes en choisissant les éléments **Changer en commande** ou **Changer en indicateur** du menu local de l'objet. La palette **Numérique** contient à la fois une commande numérique et un indicateur numérique, les deux étant fréquemment utilisés. La palette **Chaîne de caractères** contient également une commande de chaîne de caractères et un indicateur de chaîne de caractères.

L'élément de menu **Rechercher le terminal** des menus locaux de commande et d'indicateur met en évidence le terminal du diagramme correspondant à la commande ou à l'indicateur. Cet élément est utile pour identifier un objet en particulier dans un diagramme encombré.

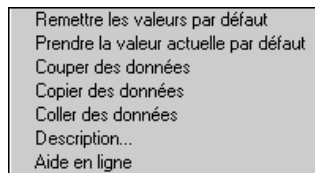
L'élément de menu **Créer»Attribute Node** crée un Attribute Node associé à l'objet. Utilisez les attribute nodes pour contrôler par un programme les différentes propriétés de l'objet.

Le sous-menu **Visualiser** affiche la liste des composantes d'une commande, telles que l'étiquette de nom et vous permet de les afficher ou de les masquer.

Lors de l'édition d'un VI, le menu local d'une commande contient un sous-menu **Opérations sur les données**. Les éléments de ce menu vous permettent de couper, copier ou coller le contenu de la commande, ramener la commande à sa valeur par défaut, définir la valeur en cours de la

commande comme nouvelle valeur par défaut, et lire ou modifier la description de la commande (lors de l'édition d'un VI, vous pouvez coller des données dans un indicateur). Certaines commandes complexes possèdent des options supplémentaires ; par exemple, une commande de type tableau permet par le biais de ses options de copier une série de valeurs et d'afficher le dernier élément du tableau.

L'illustration suivante représente le sous-menu **Opérations sur les données** d'une commande numérique en mode d'édition. Ce sous-menu représente la seule partie du menu local disponible lors de l'exécution du VI.



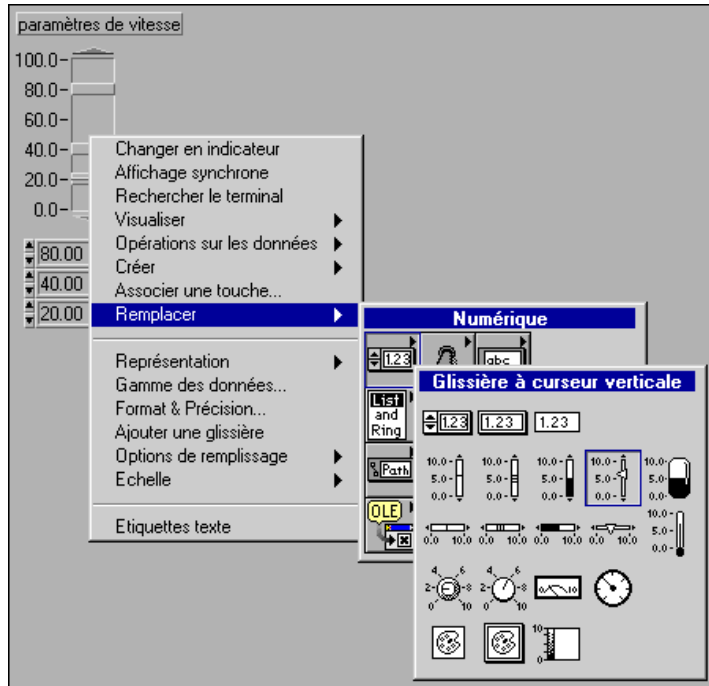
Si vous ouvrez le menu local d'une commande lors de l'exécution d'un VI, vous ne pourrez modifier que la valeur de la commande. Lors de l'exécution d'un VI, vous ne pouvez pas modifier la plupart des caractéristiques d'une commande, comme sa valeur par défaut ou sa description.

Remplacement des commandes

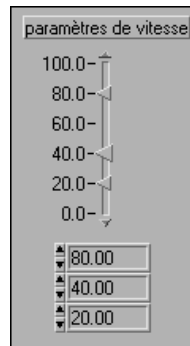
L'élément de menu **Remplacer** du menu local d'un objet affiche la palette **Commandes** à partir de laquelle vous pouvez choisir une commande ou un indicateur pour remplacer l'élément courant de la face-avant.

Utilisez **Remplacer** lorsque vous souhaitez choisir un autre style de commande mais que vous ne voulez pas perdre toutes les options d'édition jusque-là sélectionnées. En sélectionnant **Remplacer** dans le menu local, vous conservez autant d'informations que possible associées à la commande originale, telles que son nom, sa description, ses données par défaut, la direction du flux de données (commande ou indicateur), ses couleurs, sa taille, etc. Toutefois, la nouvelle commande conserve son propre type de données. Les fils de liaison du terminal de la commande ou des variables locales restent connectés dans le diagramme.

Plus la commande est similaire à la commande remplacée, plus les caractéristiques originales sont conservées. L'illustration suivante représente un exemple de glissière remplacée par un autre style de glissière.



L'illustration suivante montre le résultat.



La nouvelle glissière conserve les mêmes taille, échelle, valeur, nom, description, etc.

Par contre, si vous ouvrez un menu local et remplacez la glissière par une chaîne de caractères, uniquement le nom, la description et la direction du flux de données seront conservés car une glissière n'a que peu de points communs avec une chaîne de caractères.

Il existe une autre manière de remplacer une commande sans utiliser le menu local **Remplacer**, mais celle-ci ne conserve aucune caractéristique de la précédente commande. Elle consiste à copier la commande dans le Presse-papiers. Cette méthode maintient les connexions du diagramme et le connecteur du VI. Copiez la nouvelle commande dans le Presse-papiers. Sélectionnez ensuite l'ancienne commande que vous souhaitez remplacer avec l'outil Flèche, et sélectionnez **Edition**»**Coller**. L'ancienne commande est éliminée et la nouvelle commande est collée à sa place.

Option d'utilisation du clavier pour les commandes

Toutes les commandes de la face-avant possèdent un élément de menu **Associer une touche...** Vous pouvez l'utiliser pour associer une combinaison de touches du clavier à une commande. Lorsqu'un utilisateur appuie sur cette combinaison de touches lors de l'exécution d'un VI, le G agit comme si l'utilisateur avait cliqué sur cette commande. La commande associée devient le focus de la touche. Si c'est une commande de texte, tout le texte contenu dans cette commande est mis en évidence, prêt à être réédité. Si c'est une commande booléenne, l'état du bouton bascule.



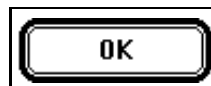
Remarque

L'élément de menu Associer une touche... est désactivé pour les indicateurs car vous ne pouvez pas saisir de données dans un indicateur.

Vous pouvez également utiliser **Associer une touche...** pour indiquer si une commande doit être prise en compte lorsque l'utilisateur passe d'une commande à une autre en utilisant la touche de tabulation pendant l'exécution.

Vous pouvez utiliser **Associer une touche...** pour associer des touches de fonction aux différents boutons qui contrôlent le comportement de la face-avant. Vous pouvez l'utiliser pour définir un bouton par défaut pour des VIs qui se comportent comme une boîte de dialogue. De cette manière, le fait d'appuyer sur la touche <Entrée> (**Windows** et **HP-UX**) ou <Retour> (**Macintosh** et **Sun**) revient à cliquer sur le bouton par défaut.

Si vous associez la touche <Entrée> ou <Retour> à un bouton de boîte de dialogue, le G dessine automatiquement ce bouton avec une épaisseur de bordure spéciale, comme dans la figure suivante.

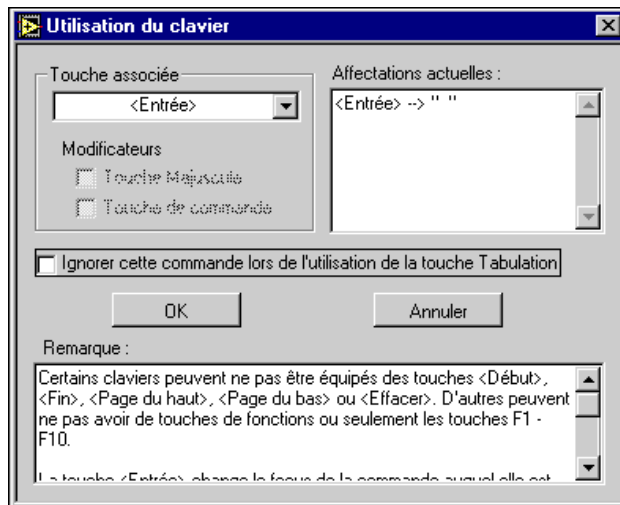


Deux points importants concernant la touche <Entrée> ou <Retour> :

1. Si vous associez la touche <Entrée> ou <Retour> à une commande, aucune commande de cette face-avant ne pourra recevoir de retour chariot. Par conséquent, toutes les chaînes de caractères de la face-avant seront limitées à une simple ligne.
2. Lorsque vous passez d'une commande à une autre en utilisant la touche de tabulation, les boutons sont traités de manière spécifique. Si vous passez à un bouton en utilisant la touche de tabulation et que vous appuyez sur la touche <Entrée> ou <Retour>, c'est le bouton sur lequel vous vous trouvez qui est actionné même si c'est une autre commande qui est associée à la touche <Entrée> ou <Retour>.

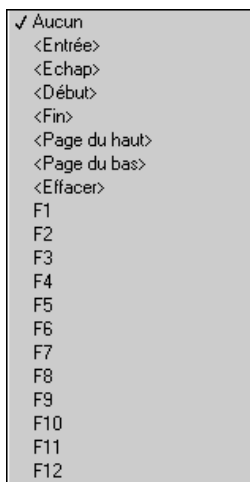
Vous ne pouvez pas attribuer la même combinaison de touches à plusieurs commandes dans une même face-avant.

Lorsque vous sélectionnez l'option **Associer une touche...**, le G affiche la boîte de dialogue suivante :



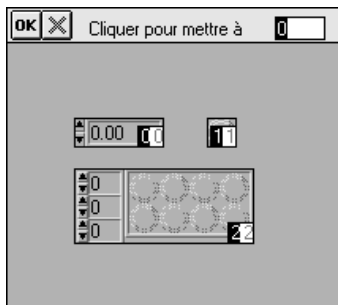
Utilisez le menu de la partie supérieure gauche de cette boîte de dialogue pour sélectionner la touche associée à cette commande. Ce peut être une simple touche ou une combinaison de touches avec la touche de modification <Maj>. En plus de la touche de modification <Maj>, vous pouvez choisir la touche <Alt> (**Windows et HP-UX**), <commande> (**Macintosh**) ou <meta> (**Sun**). Les options de ce menu sont représentées dans l'illustration suivante.

Une liste des touches du clavier actuellement associées, intitulée **Affectations actuelles** (représentée dans la précédente illustration), se situe en haut à droite de la boîte de dialogue **Utilisation du clavier**.



Option d'ordonnancement de la face-avant

Les commandes et les indicateurs d'une face-avant ont un ordre logique, appelé ordonnancement de la face-avant, et qui ne correspond pas à leur position. La première commande ou le premier indicateur que vous créez sur la face-avant est l'élément 0, le deuxième est le 1, et ainsi de suite. Si vous supprimez une commande ou un indicateur, l'ordonnancement de la face-avant s'ajuste automatiquement. Vous pouvez modifier l'ordonnancement de la face-avant en sélectionnant **Edition» Ordonnancement de la face-avant...** L'apparence de la face-avant change car elle est maintenant dans le mode d'édition de l'ordonnancement comme indiqué dans l'illustration suivante.



Les boîtes blanches des commandes ou des indicateurs affichent leurs places actuelles dans l'ordonnement de la face-avant. Les boîtes noires indiquent leur nouvelle place. En cliquant sur un élément avec le curseur d'ordonnement de la face-avant, vous pouvez définir la position de l'élément à l'aide de la commande affichée à l'intérieur de la barre d'outils. Vous pouvez modifier ce numéro en tapant un nouveau numéro à la place. Lorsque l'ordonnement de la face-avant vous convient, cliquez sur le bouton **Entrée** pour le définir et quittez le mode d'édition de l'ordonnement de la face-avant. Cliquez sur le bouton **X** pour retourner à l'ancien ordonnancement et quitter le mode d'édition de l'ordonnement de la face-avant.



Bouton X

L'ordonnement de la face-avant détermine l'ordre dans lequel les commandes et les indicateurs apparaissent dans les entrées des fichiers d'enregistrement des données produits lors de l'enregistrement de la face-avant.

Personnalisation des commandes de boîte de dialogue



menu déroulant de boîte de dialogue

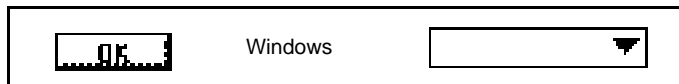


bouton de dialogue

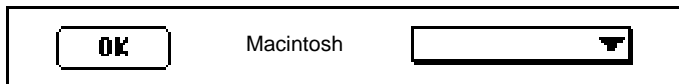
Il existe plusieurs types de commandes de boîte de dialogue : les menus déroulants (qui sont numériques), les boutons (qui sont booléens), les cases à cocher et les boutons de dialogue.

Les commandes de dialogue changent d'apparence en fonction de la plate-forme que vous utilisez. Chacune d'entre elles s'affiche avec l'apparence et la couleur spécifiques à la plate-forme.

(Windows) Le bouton de dialogue est un bouton gris, rectangulaire et en trois dimensions. Le menu déroulant est un rectangle plat en noir et blanc, comme indiqué dans l'illustration suivante.



(Macintosh) Le bouton de dialogue est un ovale noir et blanc en deux dimensions. Le menu déroulant est un rectangle noir et blanc ombré, comme indiqué dans l'illustration suivante.



(UNIX) Le bouton de dialogue est un bouton gris, rectangulaire et en trois dimensions. Le menu déroulant est un rectangle plat en noir et blanc ombré, comme indiqué dans l'illustration suivante. Vous ne pouvez pas colorer le bouton ni le menu déroulant.



Ces commandes changeant d'apparence, vous pouvez créer des VIs avec des commandes dont l'apparence est compatible avec n'importe quel ordinateur qui peut exécuter le G. En utilisant ces commandes, associées à des cases à cocher, des boutons radio booléens, des commandes numériques simples, de simples chaînes de caractères et des polices de dialogue, vous avez la possibilité de créer un VI qui adapte son apparence à l'ordinateur sur lequel il est exécuté. En utilisant les options de **Configuration du VI** pour masquer la barre de menus et les barres de défilement, et pour contrôler le comportement des fenêtres, vous pouvez créer des VIs qui ressemblent aux boîtes de dialogue standard de votre type d'ordinateur.

Personnalisation des commandes avec des graphiques importés

Vous pouvez importer des graphiques d'autres applications pour les utiliser en arrière-plan, tels que des éléments des commandes de type menu déroulant, ou des parties d'autres commandes. Pour plus d'informations sur l'utilisation d'images dans ces commandes, consultez le chapitre 13, *Commandes et indicateurs de type liste et menu déroulants* et le chapitre 24, *Commandes personnalisées et définitions de type*.

Avant de pouvoir utiliser une image, vous devez la charger dans le Presse-papiers utilisé par BridgeVIEW ou LabVIEW. Il existe une ou deux manières d'effectuer cette opération en fonction de la plate-forme en question, comme indiqué ci-dessous.

(Windows) Si vous pouvez copier un graphique directement à partir d'un programme de dessin dans le Presse-papiers de Windows et ensuite basculer vers l'application en langage G, le graphique sera automatiquement importé dans le Presse-papiers du G. Faites glisser un fichier graphique directement du système de fichier ou utilisez l'élément de menu **Importer une image à partir d'un fichier...** du menu **Edition** de Windows pour importer un fichier graphique dans le Presse-papiers du G.

Sous Windows 3.1, utilisez la dernière méthode sur les fichiers BMP, CLP, EMF, GIF, JPG, LZW, PCX, PNG, TARGA, TIFF, WMF et WPG. Sous Windows 95/NT, utilisez les fichiers BMP, CLP, EMF, JPG, PNG et WMF.

Sous Windows 95/NT, vous pouvez effectuer un glisser-déplacer avec des images bitmaps et des métafichiers directement sur les fenêtres de LabVIEW et BridgeVIEW.

(Macintosh) Si vous pouvez copier un graphique directement à partir d'un programme de dessin dans le Presse-papiers de Windows et ensuite basculer vers l'application en langage G, le graphique sera automatiquement importé dans le Presse-papiers du G.

(UNIX) Utilisez l'élément de menu **Importer une image à partir d'un fichier...** du menu **Edition** d'UNIX pour importer un graphique du type X Window Dump (XWD), que vous pouvez créer en utilisant la commande `xwd`, ou une image JPG ou PNG.

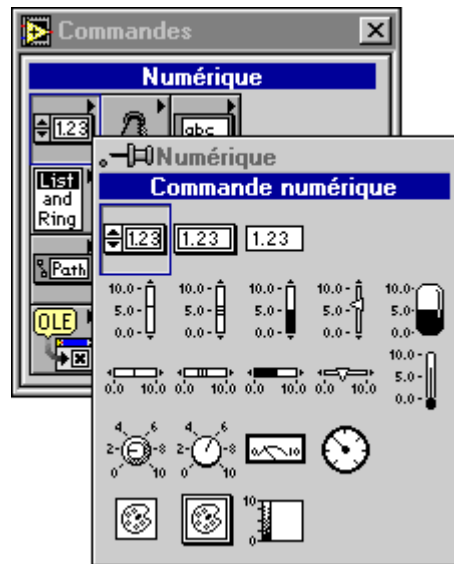
(Toutes les plates-formes) Lorsqu'un graphique se trouve dans le Presse-papiers de l'application en langage G, vous pouvez le coller comme image statique sur votre face-avant. Sinon, vous pouvez utiliser l'élément de menu **Importer une image** d'un menu local, ou encore les options de la fonction **Importer une image** de l'éditeur de commandes.

Pour obtenir un exemple de la manière d'importer des images à partir d'autres programmes, consultez le fichier `examples\general\controls\custom.llb`.

Commandes et indicateurs numériques

Ce chapitre décrit comment créer, utiliser et configurer les divers styles de commandes et indicateurs numériques.

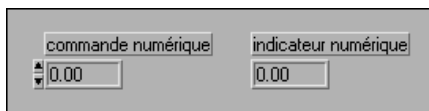
Lorsque vous sélectionnez **Commandes»Numérique** à partir de la palette, une nouvelle palette de commandes et d'indicateurs apparaît, comme le montre l'illustration suivante.



Les commandes et indicateurs numériques sont soit des commandes numériques, des commandes à glissière, des commandes rotatives, des commandes de type menu déroulant, des commandes énumérées, des commandes à boîte de couleurs ou des commandes à rampe de couleurs.

Commandes et indicateurs numériques

Une commande et un indicateur numériques sont représentés dans l'illustration suivante.



Une représentation numérique constitue la manière la plus simple d'entrer et d'afficher des données numériques.



Outil Doigt

Utilisez l'outil Doigt et l'une des méthodes suivantes pour augmenter ou diminuer la valeur affichée :

- Cliquez à l'intérieur de la fenêtre d'affichage numérique et entrez des nombres à partir du clavier.
- Cliquez sur la flèche d'incrémentacion ou de décrémentation de la commande ou de l'indicateur numérique.
- Placez le curseur à droite du chiffre que vous voulez changer et appuyez sur la flèche de déplacement vers le haut ou vers le bas sur le clavier.



Bouton Entrée

Le bouton **Entrée** apparaît sur la barre d'outils pour vous rappeler que la nouvelle valeur remplace l'ancienne uniquement lorsque vous appuyez sur la touche <Entrée> du clavier numérique, que vous cliquiez en-dehors de la fenêtre d'affichage ou que vous cliquiez sur le bouton **Entrée**. Pendant que le VI est en cours d'exécution, n'importe laquelle de ces actions empêche le logiciel d'interpréter des valeurs intermédiaires comme des entrées. Par exemple, lorsque vous changez une valeur dans l'affichage numérique pour lui affecter la valeur de 135, vous ne voulez pas que le VI reçoive les valeurs 1 et 13 avant de recevoir 135. Cette règle ne s'applique pas aux valeurs que vous changez à l'aide des flèches d'incrémentacion/décrémentacion.

Les commandes numériques acceptent seulement les entrées suivantes : des chiffres décimaux, un séparateur décimal, +, -, e en majuscules ou en minuscules, les termes Inf (infini) et NaN (not a number [pas un nombre]) et, en format de temps absolu, les caractères suivants : /, :, et am et pm en lettres majuscules ou minuscules. Si vous dépassez la limite pour la représentation sélectionnée, le logiciel ramène le nombre à sa limite naturelle. Par exemple, si vous entrez 1234 dans une commande prévue pour des entiers de type octet, le logiciel ramène le nombre à 127. Si vous

entrez de façon incorrecte des valeurs non numériques, comme aNN pour NaN ou 1fn pour Inf, l'application les ignore et utilise la valeur antérieure.

Les boutons d'incrémentement changent généralement le chiffre des unités. Augmenter un chiffre au-delà de 9 ou le diminuer en dessous de 0 affecte le chiffre adjacent approprié. Toute augmentation ou diminution est répétée automatiquement. Si vous cliquez sur un bouton d'incrémentement et que vous maintenez le bouton de la souris enfoncé, la valeur affichée augmente ou diminue de façon répétée. Si vous maintenez enfoncée la touche <Maj> pendant que vous augmentez de façon répétée la valeur, la taille de l'incrément augmente par ordres de grandeur croissants. Par exemple, par unités, puis par dizaines, par centaines, et ainsi de suite. Lorsque vous approchez de la limite de la gamme, l'ordre de grandeur de l'incrément diminue, ralentissant jusqu'à la vitesse normale à mesure que la valeur se rapproche de la limite.

Dans une commande numérique pour l'heure et la date, il est possible d'augmenter et de diminuer aussi bien les composantes individuelles (seconde, minute, heure, jour, mois, an) que les composantes individuelles de temps relatif.

Pour augmenter un affichage numérique en modifiant un chiffre autre que le chiffre des unités, utilisez l'outil Doigt pour placer le point d'insertion à droite du chiffre visé. Une fois que vous avez terminé, le chiffre des unités redevient le chiffre d'incrémentement.

Les nombres peuvent devenir trop grands pour tenir dans la zone d'affichage numérique de la commande. Vous pouvez visualiser la valeur complète en redimensionnant la commande, c'est-à-dire en augmentant sa longueur dans le sens horizontal.

Options des valeurs numériques

Vous pouvez changer les paramètres par défaut des valeurs numériques grâce à leurs menus locaux. Le menu local d'une valeur numérique est représenté dans l'illustration ci-après.

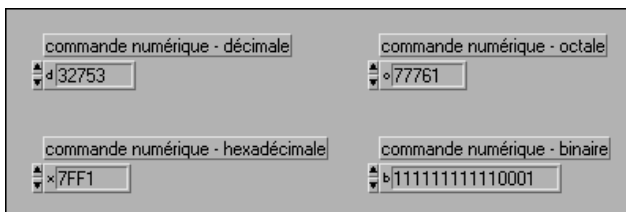


Afficher des entiers dans d'autres bases

Vous pouvez afficher des entiers signés ou non signés sous forme hexadécimale, octale et binaire, en plus de la forme décimale. Pour changer de forme, sélectionnez **Visualiser»Base** à partir du menu local de la commande ou de l'indicateur numérique. Un **d** apparaît dans l'emplacement de l'affichage numérique, comme cela est représenté dans l'illustration suivante.

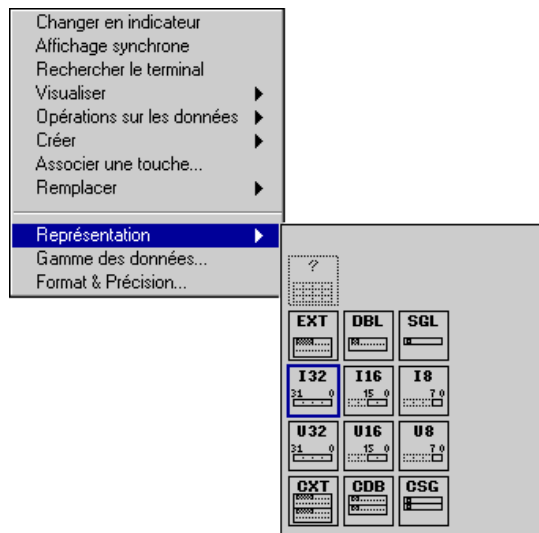


Si vous cliquez sur le **d**, le menu représenté dans l'illustration précédente apparaît. Le nombre 32 753 est affiché dans chacune des bases dans l'illustration ci-après.



Changer la représentation des valeurs numériques

Vous avez le choix entre 12 représentations pour une commande ou un indicateur numérique. Utilisez l'élément **Représentation** à partir du menu local de la commande ou de l'indicateur pour passer en simple précision (SGL) à 32 bits, double précision (DBL) à 64 bits ou précision étendue (EXT) en ce qui concerne les nombres à virgule flottante, ou pour choisir l'une des six représentations d'entiers : octet (8 bits) signé (I8) ou non signé (U8), mot (16 bits) signé (I16) ou non signé (U16) ou entiers longs (32 bits) signés (I32) ou non signés (U32). Vous pouvez aussi choisir les représentations complexe précision étendue (CXT), complexe double précision (CDB) ou complexe simple précision (CSG) pour les nombres à virgule flottante. Ces choix sont représentés dans l'illustration suivante.



Options de gamme des commandes et indicateurs numériques

Chaque représentation a des limites de gamme naturelles, minimum et maximum. Par exemple, les entiers de type octet signés sont limités à des valeurs de -128 à 127 . Les nombres à virgule flottante sont contenus dans les gammes représentées dans le tableau suivant.

Tableau 9-1. Options de gamme des nombres à virgule flottante

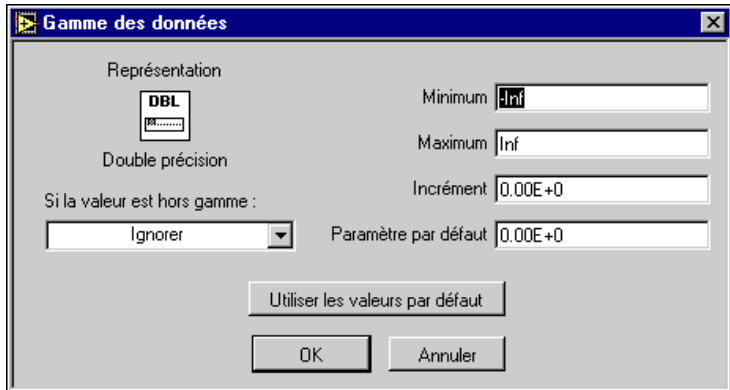
Précision	Simple	Double	Etendue (fonction de la plate-forme)
Nombre positif minimum	1.5E-45	5.0E-24	1.9E-4951
Nombre positif maximum	3.4E38	1.7E308	1.1E4932
Nombre négatif minimum	-1.5E-45	-5.0E-324	-1.9E-4951
Nombre négatif maximum	-3.4E38	-1.7E308	-1.1E4932



Remarque

Bien que le G puisse effectuer des calculs dans la gamme indiquée dans le tableau 9-1, Options de gamme des nombres à virgule flottante, la gamme des nombres à virgule flottante en précision étendue qu'il peut représenter et afficher en format texte est $\pm 9.9999999999999999E999$. Pour le HP-UX, le Concurrent et le Power Macintosh, la gamme pour les nombres à virgule flottante en précision étendue est la même que pour les nombres en double précision.

Vous pouvez choisir d'autres limites contenues dans ces domaines naturels grâce à l'élément **Gamme des données...** du menu local. La boîte de dialogue suivante apparaît.



Contrôle des gammes numériques

Vous pouvez aussi astreindre des valeurs intermédiaires à certains incréments. Par exemple, vous pouvez limiter les entiers mots à des incréments de 10, ou les nombres à virgule flottante en simple précision à des incréments de 0,25. Si vous changez les limites ou l'incrément, vous devez décider de la marche à suivre si un VI ou l'opérateur essaie de fixer une valeur en dehors de la gamme ou ne correspondant pas à l'incrément. Vous avez les options suivantes :

Ignorer

Le G ne modifie pas et n'indique pas les valeurs incorrectes. En cliquant sur les flèches d'incrément et de décrémentation, vous ajoutez ou soustrayez à la valeur l'incrément que vous avez défini, mais la valeur ne dépasse pas les valeurs minimum ou maximum.

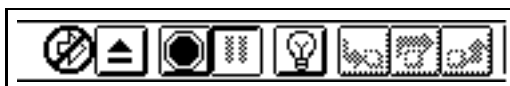
Contraindre

Le G ramène automatiquement les valeurs incorrectes à la valeur valide la plus proche. Par exemple, si le minimum est de 3, le maximum est de 10 et l'incrément est de 2, les valeurs valides sont 3, 5, 7, 9 et 10. Le G contraint la valeur 0 à 3, la valeur 6 à 7 et la valeur 100 à 10.

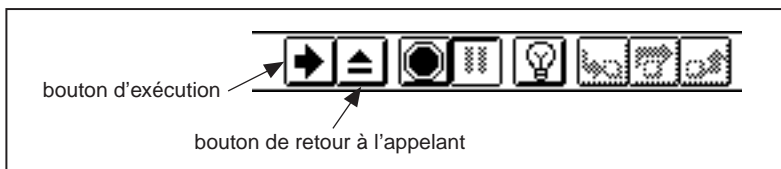
Interrompre

Le G interrompt l'exécution d'un VI ou d'un sous-VI lorsqu'une valeur est incorrecte. Si vous choisissez d'interrompre en cas de valeur incorrecte, le G garde en mémoire une copie de toutes les données de la face-avant, au cas où vous auriez besoin d'ouvrir la face-avant pour mettre en évidence une erreur.

Un VI principal avec des commandes dont les valeurs sont incorrectes ne peut pas s'exécuter. Si la valeur d'une commande est incorrecte avant l'exécution d'un sous-VI (lorsqu'un sous-VI est sur le point de s'exécuter), le VI est interrompu. La face-avant du VI s'ouvre (ou devient la fenêtre active) et les commandes incorrectes sont encadrées en rouge (ou par une épaisse ligne noire sur un moniteur noir et blanc). La barre d'outils sur le diagramme d'un sous-VI interrompu correspond à l'illustration suivante.



Vous devez ramener la commande à une valeur valide avant de pouvoir continuer. Lorsque les valeurs de toutes les commandes sont valides, la barre d'outils correspond à l'illustration suivante. Cliquez sur le bouton **Exécution** pour poursuivre l'exécution.



Si la valeur d'un indicateur ou d'une commande devient incorrecte pendant l'exécution d'un sous-VI, l'exécution s'arrête comme s'il y avait un point d'arrêt. La face-avant du VI s'ouvre (ou devient la fenêtre active). Les indicateurs et commandes alors incorrects sont encadrés en rouge (ou par une épaisse ligne noire sur un moniteur noir et blanc). Lorsqu'un VI suspendu finit son exécution, s'il y a des valeurs incorrectes, le bouton **Retour au VI appelant** est désactivé, comme cela est représenté dans l'illustration suivante.



Vous devez ramener les indicateurs à des valeurs valides avant de pouvoir retourner au VI appelant. Vous pouvez également changer les valeurs des commandes pour produire des sorties valides et exécuter à nouveau le sous-VI en cliquant sur le bouton **Exécution**. Lorsque les valeurs de tous les indicateurs sont valides, la barre d'outils correspond à l'illustration suivante. Cliquez sur le bouton **Retour au VI appelant** pour poursuivre l'exécution.



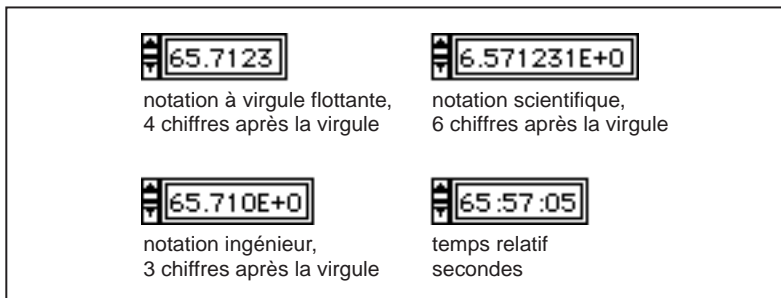
Format et précision des affichages numériques

Vous pouvez sélectionner le format de vos affichages numériques pour les valeurs numériques ou pour l'heure et la date. S'il s'agit d'une valeur numérique, vous avez le choix entre notation en virgule flottante, notation scientifique, notation ingénieur ou temps relatif en secondes. Vous pouvez sélectionner leur précision, c'est-à-dire le nombre de chiffres qui sont affichés après la virgule, de 0 à 20. La précision que vous sélectionnez affecte seulement l'affichage de la valeur ; la précision interne dépend toujours de la représentation.

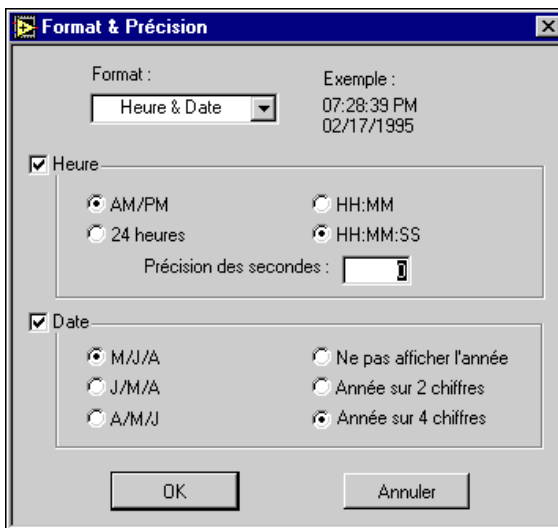
Pour changer des paramètres de l'affichage numérique, sélectionnez l'élément **Format & Précision...** du menu local de l'affichage. La boîte de dialogue s'ouvre en format numérique, comme cela est représenté dans l'illustration suivante.



Vous pouvez voir un exemple en haut de la boîte de dialogue pendant que vous faites vos sélections. Des exemples de réglages de format et de précision sur une commande numérique sont représentés dans l'illustration suivante.



Pour formater le temps absolu et/ou la date, sélectionnez **Heure & Date** à partir du menu déroulant Format qui se trouve en haut de la boîte de dialogue. La boîte de dialogue change, comme cela est représenté dans l'illustration suivante.



Vous pouvez formater l'heure ou la date, ou les deux. Si vous entrez uniquement l'heure ou uniquement la date, les composantes non spécifiées sont déduites. Si vous n'entrez pas l'heure, l'heure adoptée par défaut est minuit. Si vous n'entrez pas la date, la valeur précédente de la date est adoptée. Si vous entrez une date, mais que la commande n'est pas dans un format de date, l'ordre mois, jour et année adopté est celui qui correspond

aux paramètres établis dans la boîte de dialogue Préférences. Si vous entrez seulement deux chiffres pour l'année, il est supposé que tout nombre inférieur à 38 appartient au XXI^{ème} siècle, et que tout nombre supérieur ou égal à 38 appartient au XX^{ème} siècle.

Bien que le temps absolu soit affiché sous la forme d'une chaîne de caractères contenant l'heure et la date, il est représenté à l'intérieur du système par le nombre de secondes écoulées depuis le 1 janvier 1904 à minuit, "Temps universel coordonné" (Universal Coordinated Time [UTC]), anciennement appelé "Temps moyen de Greenwich" (Greenwich Mean Time [GMT]). Le logiciel garde ces composantes à jour à l'intérieur du système.



Remarque

Lorsqu'une commande est en format de temps absolu, vous avez toujours la possibilité d'entrer l'heure, la date, ou l'heure et la date. Si vous ne voulez pas qu'une date soit adoptée par défaut, utilisez le temps relatif.

Notez les exemples en haut à droite de la boîte de dialogue, qui changent lorsque vous faites votre sélection.

La gamme acceptable pour l'heure et la date varie d'une plate-forme informatique à une autre, comme suit :

(Windows) 2 janvier 1970, 0h00 – 4 février 2106, 0h00

(Windows 95/NT) 2 janvier 1970, 0h00 – 17 janvier 2038, 0h00

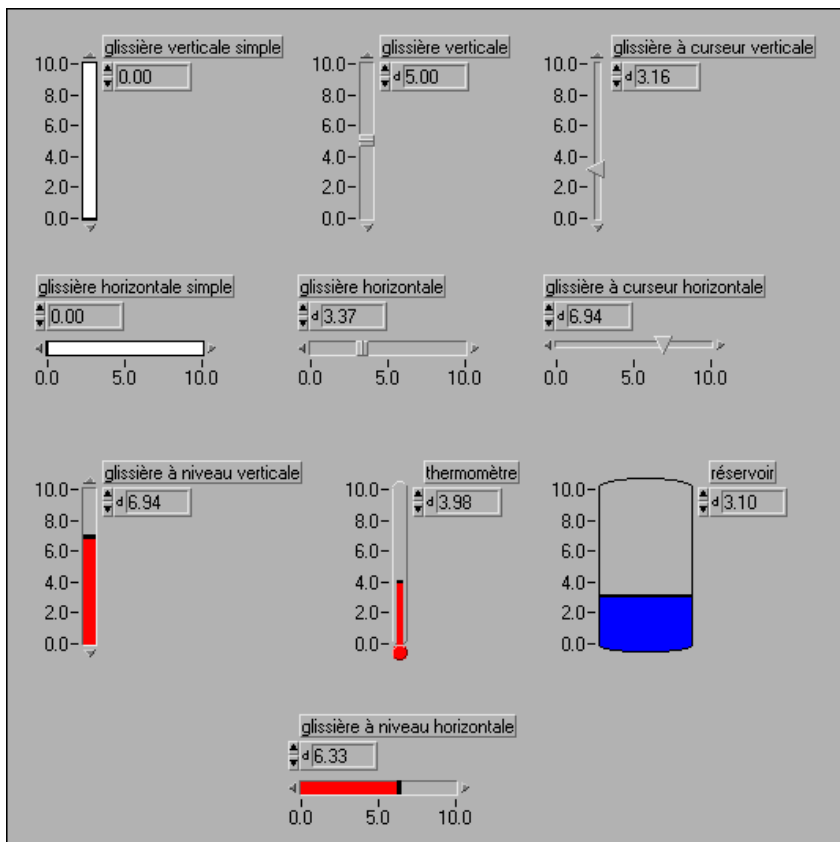
(Macintosh) 2 janvier 1904, 0h00 – 2 janvier 2040, 0h00

(UNIX) 15 décembre 1901, 0h00 – 17 janvier 2038, 0h00

Ces gammes peuvent être jusqu'à quelques jours plus larges, suivant le fuseau horaire dans lequel vous vous trouvez et si l'heure d'été est en vigueur.

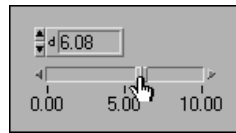
Commandes et indicateurs numériques à glissière

Les commandes et indicateurs à glissière sont représentés dans l'illustration ci-après.



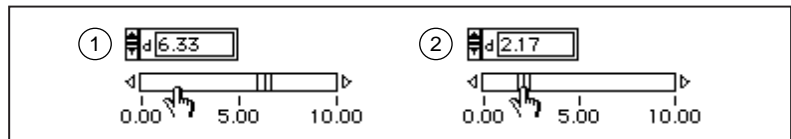
Chaque glissière a un affichage numérique. Utilisez les affichages numériques pour entrer des données dans les commandes à glissière, comme cela est expliqué dans la section *Commandes et indicateurs numériques* de ce chapitre. Utilisez l'outil Doigt sur les parties telles que la glissière, l'emplacement de la glissière, l'échelle et les boutons d'incrémentación pour entrer des données ou changer des valeurs. La glissière est la partie qui se déplace pour indiquer la valeur de la commande. L'emplacement est la partie stationnaire sur laquelle se déplace la glissière. L'échelle indique la valeur de la glissière, et les boutons d'incrémentación

sont des petits triangles situés à chaque extrémité du boîtier. Un exemple de glissière est représenté dans l'illustration suivante.

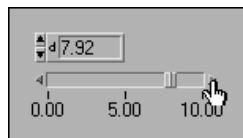


À l'aide de l'outil Doigt, vous pouvez faire glisser la glissière jusqu'à une nouvelle position. Si le VI est en cours d'exécution au moment du changement, des valeurs intermédiaires peuvent être transmises au programme, suivant la fréquence à laquelle le VI lit la commande.

Vous pouvez aussi cliquer sur un point de l'emplacement et la glissière effectue un saut jusqu'à cet endroit, comme cela est représenté dans l'illustration suivante. Aucune valeur intermédiaire n'est transmise au programme.



Si vous utilisez une glissière qui a des boutons d'incrémentations, vous pouvez cliquer sur un bouton d'incrémentations, comme cela est représenté dans l'illustration suivante, et la glissière se déplace lentement dans la direction de la flèche. Maintenez enfoncée la touche <Maj> pour déplacer la glissière plus rapidement. Des valeurs intermédiaires peuvent être transmises au programme.



Tout comme les affichages numériques, les glissières ont des options **Représentation**, **Gamme des données...** et **Format & Précision** dans leurs menus locaux. Ces options fonctionnent de la même manière que pour les affichages numériques, à ceci près que les glissières ne peuvent pas

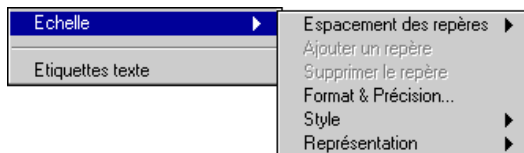
représenter les nombres complexes. Les glissières ont aussi d'autres options. L'illustration suivante montre un menu local de glissière.



L'élément de menu **Visualiser**»**Affichage numérique** du menu local contrôle la visibilité de l'affichage numérique de la glissière.

Echelle de la glissière

Les éléments du sous-menu d'échelle s'appliquent uniquement à l'échelle. Le menu local d'échelle est représenté dans l'illustration suivante.



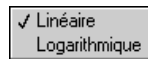
L'élément **Espacement des repères** est décrit dans la section *Repères d'échelle* de ce chapitre.

L'élément **Format & Précision** fonctionne de la manière décrite dans la section *Format et précision des affichages numériques* de ce chapitre.

L'élément **Style** affiche la palette représentée dans l'illustration suivante. Vous pouvez afficher une échelle sans graduation ni valeurs d'échelle, ou vous pouvez masquer complètement l'échelle.



L'élément **Représentation** vous offre une option d'espacement linéaire ou logarithmique de l'échelle, comme cela est représenté dans l'illustration suivante.



Si vous passez en espacement logarithmique, et si la limite inférieure de l'échelle est inférieure ou égale à 0, la limite devient automatiquement un nombre positif, et les autres marqueurs sont réévalués en conséquence. N'oubliez pas que les options d'échelle, y compris les fonctions de représentation, sont indépendantes des valeurs de gamme données de la glissière, que vous pouvez changer avec l'option **Gamme des données** du menu local. Si vous voulez limiter vos données à des valeurs logarithmiques, changez la gamme des données pour éliminer les valeurs inférieures ou égales à zéro.

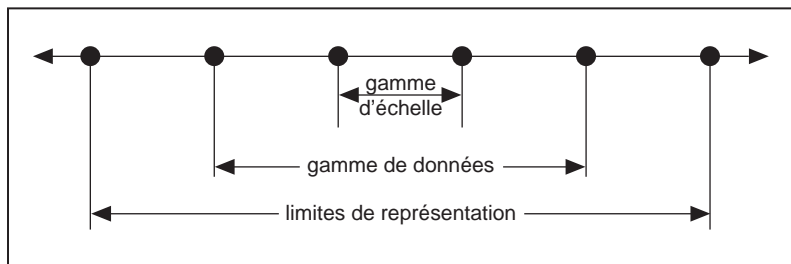
Repères d'échelle

L'échelle d'une commande ou d'un indicateur numérique a au moins deux repères, qui sont des étiquettes indiquant la valeur associée à la position du repère.

Changer les limites d'une échelle

Les deux repères les plus extrêmes sont les limites de l'échelle. Il n'est pas nécessaire qu'ils coïncident avec les limites de la gamme, mais ils peuvent être un sous-ensemble de la gamme de la commande ou de l'indicateur. Par exemple, la gamme par défaut d'une commande ou d'un indicateur à entier signé de 16 bits va de -32 768 à 32 767. Cependant, vous pouvez définir la

gamme de données de -1 000 à 1 000, puis fixer les limites de l'échelle à 0 et 500.



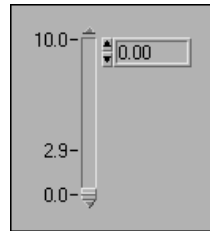
Vous pouvez changer interactivement la valeur minimale, la valeur maximale et l'incrément d'une échelle de cinq manières différentes à l'aide de l'outil Doigt ou de l'outil Texte. Si vous voulez changer une échelle au moyen d'un programme, utilisez un attribut node. Pour plus d'informations sur la façon de changer une échelle au moyen d'un programme grâce à un attribut node, consultez le chapitre 22, *Attribute Nodes*.

- Si vous tapez une nouvelle valeur maximale dans sa zone d'affichage, la valeur minimale reste la même, et le G recalcule automatiquement l'incrément.
- Si vous tapez une nouvelle valeur minimale dans sa zone d'affichage, la valeur maximale reste la même, et le G recalcule automatiquement l'incrément.
- Si vous tapez la valeur minimale actuelle dans la zone d'affichage de la valeur maximale, le G inverse l'échelle, de sorte que ce qui était auparavant la valeur minimale devient maintenant la valeur maximale, et ce qui était auparavant la valeur maximale devient maintenant la valeur minimale. Le G recalcule également les incréments.
- Si vous tapez une valeur dans un repère intermédiaire, l'incrément devient la différence entre cette valeur et la valeur minimale.
- Si vous changez la taille de la glissière, l'incrément s'ajuste pour que les repères ne se chevauchent pas.

Sélectionner une distribution non uniforme de repères d'échelle

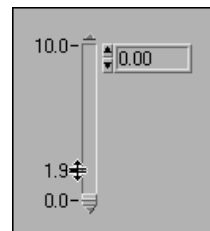
Par défaut, les repères d'échelle sont régulièrement espacés. Si vous voulez, vous pouvez indiquer exactement où doivent être situés les repères intermédiaires pour une échelle. Ceci est utile pour repérer quelques points spécifiques sur une glissière (comme un point de consigne ou un seuil). Si

vous voulez une distribution non uniforme des repères, sélectionnez **Echelle»Espacement des repères»Arbitraire** à partir du menu local de l'échelle. Ensuite, vous pouvez ouvrir le menu local de la glissière et sélectionner **Echelle»Ajouter un repère** ou ouvrir le menu local du repère et sélectionner **Ajouter un repère**. Un repère apparaît dans une position arbitraire, comme cela est représenté dans l'illustration suivante.



Pour supprimer un repère arbitraire, ouvrez le menu local de la glissière et sélectionnez **Echelle»Supprimer le repère** ou ouvrez le menu local du repère et sélectionnez **Supprimer le repère**.

Lorsqu'un repère est créé, vous pouvez taper un nombre dans le repère ; son emplacement change automatiquement pour correspondre au nombre que vous avez entré. Vous pouvez aussi déplacer des repères en faisant glisser les tirets, c'est-à-dire les petits traits de graduation qui se trouvent à côté des points sur l'échelle. Pour ce faire, maintenez l'outil Doigt inactif au-dessus des tirets. Le curseur change d'aspect pour montrer que les tirets peuvent bouger, comme cela est représenté dans l'illustration suivante.



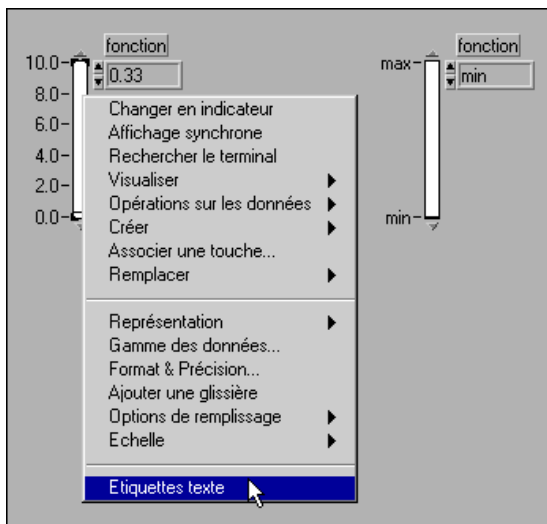
En mode Uniforme, le fait de glisser un tiret change la distribution des repères. En mode Repères arbitraires, le fait de glisser un tiret déplace seulement le repère en question, sans changer les autres repères. Si vous appuyez sur la touche **<Ctrl> (Windows)**, **<option> (Macintosh)**, **<meta> (Sun)** ou **<Alt> (HP-UX)** pendant que vous faites glisser un tiret, vous créez un nouveau repère (si les tirets sont masqués, comme choisi dans la boîte de dialogue **Echelle»Style**, vous ne pouvez pas les faire glisser).

Les repères arbitraires affectent seulement les repères intermédiaires, mais pas les deux repères extrêmes. Si aucun repère arbitraire n'est visible dans la gamme courante, l'échelle revient temporairement à des repères uniformes.

Echelle du texte

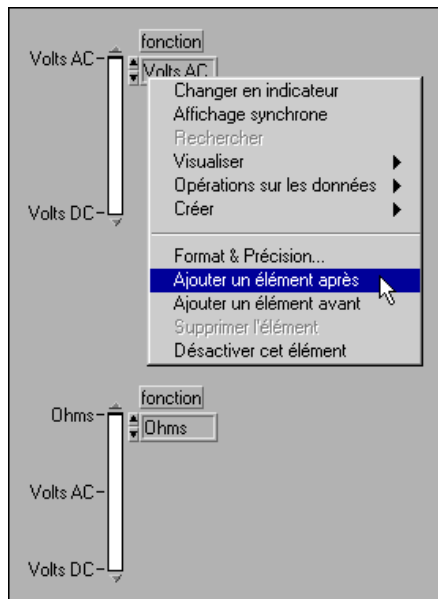
Vous pouvez aussi utiliser des étiquettes de texte sur les échelles numériques. Les étiquettes sont utiles parce que vous pouvez associer des chaînes de caractères à des valeurs entières. Cette configuration est utile pour sélectionner des options mutuellement exclusives. Pour mettre en œuvre cette option, sélectionnez **Étiquettes texte** à partir du menu local de la glissière. La commande à glissière apparaît avec les étiquettes de texte par défaut, min et max, et vous pouvez entrer immédiatement les étiquettes que vous voulez. Appuyez sur <Maj-Entrée> (**Windows et HP-UX**) ou <Maj-Retour> (**Macintosh et Sun**) après chaque étiquette pour créer une nouvelle étiquette, et sur <Entrée> sur le pavé numérique une fois que vous avez terminé d'éditer votre dernière étiquette.

Vous pouvez utiliser l'outil Texte pour éditer les étiquettes min et max dans la zone d'affichage de texte ou directement sur la glissière.



Vous pouvez ouvrir le menu local de la zone d'affichage de texte et sélectionner **Visualiser»Affichage numérique** pour savoir quelles valeurs numériques sont associées aux étiquettes de texte que vous créez. Ces valeurs commencent toujours à zéro (en bas des glissières verticales et à gauche ou à droite des glissières horizontales) et augmentent d'une unité pour chaque étiquette de texte.

Utilisez l'élément **Ajouter un élément après** ou **Ajouter un élément avant** du menu local de la zone d'affichage de texte pour créer de nouvelles étiquettes, comme cela est représenté dans l'illustration suivante.



Vous pouvez aussi appuyer sur <Maj-Entrée> pour créer un nouvel élément lorsque vous éditez les éléments existants.

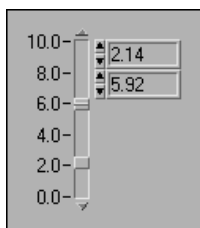
Glissières pleines et à valeurs multiples

La palette **Numérique** contient quatre commandes configurées pour effectuer un remplissage de la valeur minimale à la valeur de la glissière. Ces commandes comprennent une glissière verticale, une glissière horizontale, le réservoir et le thermomètre. Grâce à l'élément **Options de remplissage** du menu local, vous pouvez transformer toutes les glissières en glissières à niveau, et vous pouvez transformer les glissières à niveau en glissières ordinaires. Normalement, vous avez le choix entre trois

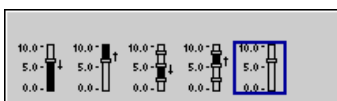
configurations, comme cela est représenté dans l'illustration suivante. La première consiste à colorier depuis la valeur minimale jusqu'à la position de la glissière, la deuxième à colorier depuis la valeur maximale jusqu'à la position de la glissière, et la troisième à ne pas colorier du tout.



Vous pouvez aussi afficher plusieurs valeurs sur la même glissière. Pour ce faire, choisissez **Ajouter une glissière** dans le menu local. Une nouvelle glissière apparaît avec un nouvel affichage numérique, comme cela est représenté dans l'illustration suivante.



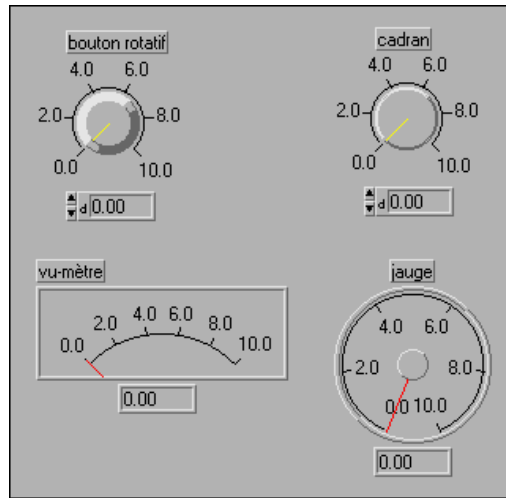
Lorsque vous faites cela, deux options supplémentaires apparaissent dans la palette **Options de remplissage** : **Remplir jusqu'à la valeur ci-dessus** et **Remplir jusqu'à la valeur ci-dessous**. Ces options s'appliquent à la glissière active. Les cinq options sont représentées dans l'illustration suivante.



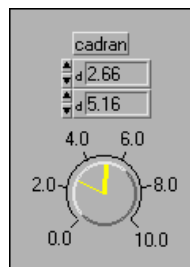
Pour un exemple des commandes et indicateurs à glissière, consultez `exemples\general\controls\alarms1d.llb`.

Commandes et indicateurs numériques rotatifs

Les commandes et indicateurs numériques rotatifs sont représentés dans l'illustration suivante.



Les objets rotatifs ont, dans l'ensemble, les mêmes options que les objets à glissière. Les glissières (ou aiguilles) des objets rotatifs tournent au lieu de glisser, mais vous pouvez les utiliser de la même manière que les glissières.

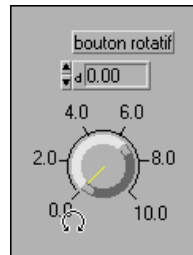


Les commandes rotatives, comme les commandes linéaires, peuvent afficher plusieurs valeurs, comme cela est représenté dans l'illustration précédente. Vous pouvez ajouter de nouvelles valeurs en sélectionnant **Ajouter une aiguille** à partir du menu local, de la même manière que vous ajoutez de nouvelles valeurs à des commandes à glissière.

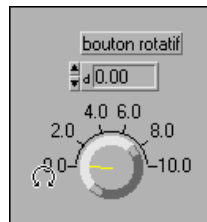
Si vous amenez l'outil Flèche sur l'échelle, l'outil se transforme en curseur rotatif. Si vous cliquez sur les limites extrêmes de l'échelle et que vous les faites glisser, vous pouvez changer l'arc décrit par l'échelle. Si vous cliquez

sur les repères intermédiaires de l'échelle et que vous les faites glisser, vous pouvez faire tourner l'arc. L'échelle a toujours la même gamme, mais les angles de ses extrémités sont différents. Cette procédure est représentée dans l'illustration suivante. Maintenez enfoncée la touche <Ma j> pour astreindre l'arc à des multiples de 45 degrés.

Placez le curseur sur l'échelle du bouton rotatif. Elle change d'apparence pour ressembler à un fer à cheval avec une flèche à chaque extrémité.



En déplaçant les repères extrêmes, vous changez la taille de l'arc de l'échelle.



En déplaçant les repères intermédiaires, vous faites tourner l'arc.



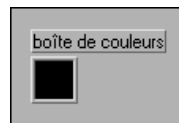
Les tirets de graduation des échelles rotatives peuvent être déplacés au moyen de l'outil Doigt, tout comme les échelles linéaires. Les tirets peuvent aussi être répartis de façon non uniforme, comme dans le cas des échelles linéaires. Pour plus d'informations, consultez la section

Sélectionner une distribution non uniforme de repères d'échelle plus haut dans ce chapitre.

Pour plus d'informations sur les commandes et indicateurs de type menu déroulant, consultez le chapitre 13, *Commandes et indicateurs de type liste et menu déroulants*.

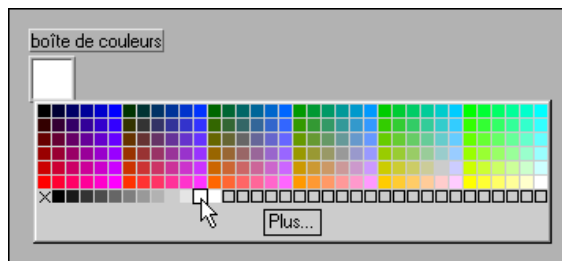
Boîte de couleurs

La boîte de couleurs affiche une couleur correspondant à une valeur spécifiée. La valeur de la couleur est exprimée en tant que nombre hexadécimal sous la forme `RRGGBB`, où les deux premiers chiffres contrôlent la valeur de la couleur rouge, les deux chiffres du milieu contrôlent la valeur de la couleur verte, et les deux derniers chiffres contrôlent la valeur de la couleur bleue. Un exemple est représenté dans l'illustration suivante.



Définissez la couleur de la boîte de couleurs en cliquant dessus avec l'outil Doigt ou avec l'outil Pinceau pour afficher une palette **Couleur**, comme

cela est représenté dans l'illustration suivante. En relâchant le bouton de votre souris sur la couleur que vous voulez, vous sélectionnez cette couleur.



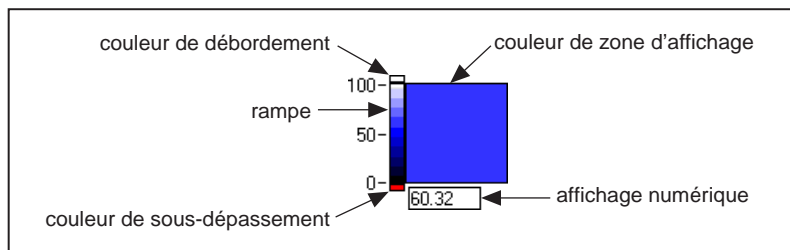
Les boîtes de couleurs sont généralement utilisées sur la face-avant en tant qu'indicateurs. La façon la plus facile de définir la couleur d'une boîte de couleurs est d'utiliser la constante boîtes de couleurs, à partir de la palette **Fonctions»Numérique»Constantes numériques supplémentaires** dans la fenêtre du diagramme. Câblez la constante boîte de couleurs au terminal de la boîte de couleurs sur le diagramme. En cliquant sur la constante boîte de couleurs avec l'outil Doigt ou avec l'outil Pinceau, vous affichez la

palette **Couleur** utilisée également pour définir la couleur de la boîte de couleurs. Si vous voulez qu'un indicateur de boîte de couleurs de la face-avant change de couleur pour indiquer différentes conditions, vous pouvez utiliser une série de constantes boîtes de couleurs à l'intérieur d'une structure Condition.

Le petit T dans le coin inférieur gauche de la palette **Couleur** représente la couleur transparente.

Rampe de couleurs

La rampe de couleurs utilise une couleur pour afficher une valeur numérique. Vous pouvez configurer une échelle de couleurs, constituée d'au moins deux repères arbitraires comportant chacun une valeur numérique et une couleur correspondant à cette valeur. Lorsque la valeur d'entrée change, la couleur affichée change pour correspondre à la nouvelle valeur. Un exemple est représenté dans l'illustration suivante.



Vous créez ces paires en utilisant les repères arbitraires de l'échelle des couleurs. Chaque repère arbitraire spécifie une valeur et a une couleur associée. Vous changez la couleur associée à un repère en ouvrant le menu local du repère, pas celui de la rampe. Vous pouvez ajouter des repères, changer les valeurs et définir les couleurs comme vous le désirez.

Utilisez l'élément **Interpoler les couleurs** du menu local de la rampe de couleurs pour déterminer si des teintes intermédiaires doivent être affichées pour les valeurs situées entre les repères indiqués. Si **Interpoler les couleurs** est désactivé, la couleur correspond au niveau le plus élevé inférieur ou égal à la valeur en question.

Le tableau de couleurs est toujours classé par valeurs de repères. L'échelle de la rampe correspond à la plus grande et à la plus petite valeur du tableau. Lorsque la valeur minimale ou la valeur maximale de l'échelle change, les niveaux du tableau de couleurs sont automatiquement redistribués entre les nouvelles valeurs.

Les rampes de couleurs disposent de plusieurs options, que vous pouvez afficher ou masquer. Ces options comprennent l'étiquette d'unité, l'affichage numérique et la rampe.

La rampe de ce type de commande a une couleur supplémentaire en haut et une autre en bas de l'échelle. Utilisez celles-ci pour sélectionner une couleur à afficher en cas de débordement par excès ou par défaut, c'est-à-dire, lorsque la valeur de la commande est supérieure à la valeur maximale ou inférieure à la valeur minimale de l'échelle des couleurs. Cliquez sur ces zones avec l'outil Doigt et sélectionnez vos couleurs de débordement par excès et par défaut à partir de la palette **Couleur**.

La rampe de couleurs et la boîte de couleurs sont toutes deux utilisées pour afficher une valeur numérique sous la forme d'une couleur. Avec la rampe de couleurs, vous pouvez affecter n'importe quelle gamme de nombres à n'importe quelle gamme de couleurs. La boîte de couleurs, par contre, affiche seulement la couleur spécifiée par les composantes rouge, verte et bleue de la valeur numérique ; une valeur donnée donne toujours lieu à la même couleur.



Remarque

La rampe de couleurs peut avoir au maximum autant de couleurs que votre moniteur peut en afficher.

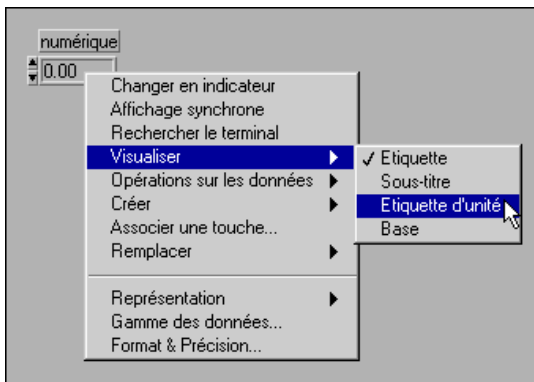
Vous pouvez utiliser l'échelle des couleurs pour créer des tables de couleurs pour les commandes Graphe d'intensité et Graphe déroulant d'intensité. Pour plus d'informations sur ces commandes, consultez le chapitre 15, *Indicateurs et commandes de graphe et de graphe déroulant*.

Types d'unités

Toute commande numérique peut avoir des unités physiques, comme des mètres ou des kilomètres/seconde, qui lui sont associées. Toute commande numérique à laquelle est associée une unité est limitée, en termes de type de données, à des nombres à virgule flottante.

Les unités affectées à une commande sont affichées et peuvent être modifiées dans une étiquette séparée mais jointe, appelée étiquette d'unité.

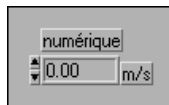
Vous pouvez visualiser cette étiquette en sélectionnant cet élément à partir du menu local, comme cela est représenté dans l'illustration suivante.



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

Si vous ne savez pas quelles sont les unités valides, entrez une unité simple, par exemple *m*, puis ouvrez le menu local de l'étiquette d'unité et sélectionnez **Unité...** Une boîte de dialogue apparaît, contenant des informations sur les unités disponibles. Vous pouvez utiliser cette boîte de dialogue pour remplacer votre première unité par un choix plus approprié.

La commande numérique suivante est configurée pour avoir comme unités des mètres par seconde.



Les tableaux 9-2 à 9-6 dressent la liste de toutes les unités que vous pouvez utiliser avec la fonction de sélection des unités.

Tableau 9-2. Unités de base

Nom de la quantité	Unité	Abréviation
angle plan	radian	rad
angle solide	stéradian	sr
temps	seconde	s
longueur	mètre	m
masse	gramme	g
courant électrique	ampère	A
température thermodynamique	kelvin	K
quantité de substance	mole	mol
intensité lumineuse	candela	cd

Tableau 9-3. Unités dérivées à noms spéciaux

Nom de la quantité	Unité	Abréviation
fréquence	hertz	Hz
force	newton	N
pression	pascal	Pa
énergie	joule	J
puissance	watt	W
charge électrique	coulomb	C
potentiel électrique	volt	V
capacité	farad	F
résistance électrique	ohm	Ohm
conductance	siemens	S
flux magnétique	weber	Wb
densité de flux magnétique	tesla	T

Tableau 9-3. Unités dérivées à noms spéciaux (Suite)

Nom de la quantité	Unité	Abréviation
inductance	henry	H
flux lumineux	lumen	lm
éclairage	lux	lx
température Celsius	degré Celsius	degC
activité	becquerel	Bq
dose absorbée	gray	Gy
dose équivalente	sievert	Sv

Tableau 9-4. Unités supplémentaires en usage avec des unités SI

Nom de la quantité	Unité	Abréviation
temps	minute	min
temps	heure	h
temps	jour	j
angle plan	degré	deg
angle plan	minute	'
angle plan	seconde	"
volume	litre	l
masse	tonne métrique	t
surface	hectare	ha
énergie	électron-volt	eV
masse	unité de masse atomique unifiée	u

Tableau 9-5. Unités CGS

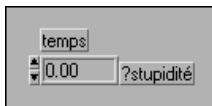
Nom de la quantité	Unité	Abréviation
surface	barn	b
force	dyne	dyn
énergie	erg	erg
pression	bar	bar

Tableau 9-6. Autres unités

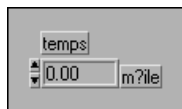
Nom de la quantité	Unité	Abréviation
température Fahrenheit	degré Fahrenheit	degF
différence de température Celsius	degré Celsius	Cdeg
différence de température Fahrenheit	degré Fahrenheit	Fdeg
longueur	pied	ft
longueur	pouce	in
longueur	mile	mi
surface	acre	acre
pression	atmosphère	atm
énergie	calorie	cal
énergie	unité thermique britannique	Btu

Entrée des unités

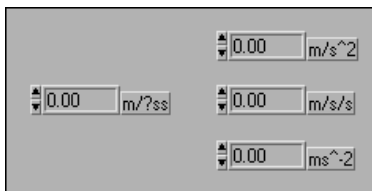
Si vous essayez d'entrer une unité incorrecte dans une étiquette d'unité, le G signale que l'unité est incorrecte en plaçant un ? dans l'étiquette, comme cela est représenté dans l'illustration suivante.



Notez que vous devez entrer des unités en utilisant les abréviations correctes, sinon le G le signale, comme cela est illustré dans l'illustration suivante.



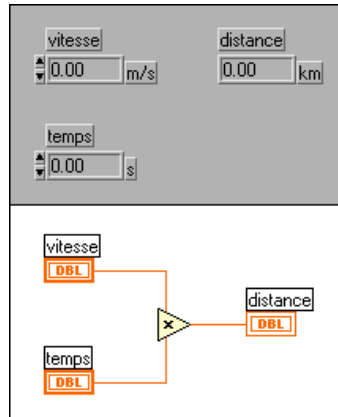
De plus, n'oubliez pas que vous ne pouvez pas entrer d'unités ambiguës. Par exemple, un indicateur d'invalidité figure en face de m/ss dans l'illustration suivante, parce qu'il n'est pas clair si cela signifie mètres par seconde carrée, ou (mètres/secondes) * secondes. Pour lever l'ambiguïté, vous devez entrer les unités de manière appropriée, comme cela est représenté dans les trois exemples suivants.



Vous ne pouvez pas sélectionner d'unités pour un graphe ou un graphe déroulant tant que vous ne les câblez pas à un objet auquel est associée une unité. Une fois le câblage réalisé, vous pouvez visualiser l'unité que le graphe ou le graphe déroulant a acquise par l'intermédiaire de la connexion. Vous pouvez la changer, mais uniquement si vous la remplacez par une unité compatible qui mesure le même phénomène physique. Par exemple, si une entrée pour un graphe déroulant est exprimée en mi, vous pouvez changer l'unité du graphe déroulant en ft, in, ou m, mais pas en N, Hz, min, acre, ou A.

Unités et contrôle de type strict

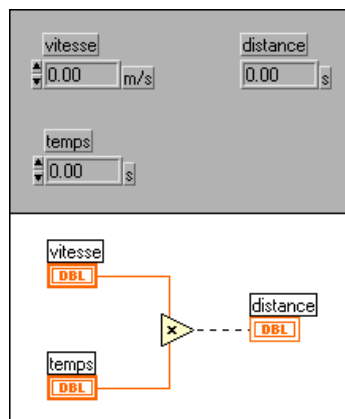
Un fil de liaison branché à une source à laquelle est associée une unité ne peut être connecté qu'à une destination dont l'unité est compatible, comme cela est représenté dans l'illustration suivante.



Dans le cas de l'illustration précédente, l'affichage de distance est automatiquement mis à l'échelle pour afficher des kilomètres au lieu des mètres, étant donné que c'est l'unité qui est spécifiée pour l'indicateur.

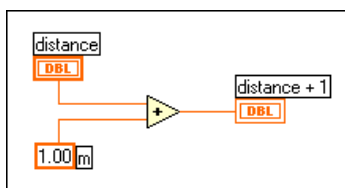
Vous ne pouvez pas connecter des signaux avec des unités incompatibles, comme cela est représenté dans l'illustration suivante. Si vous sélectionnez **Liste des erreurs** à partir du menu local du fil de liaison brisé représenté

ci-dessous, la fenêtre de Liste des erreurs affiche le message `Fil de liaison : conflit d'unité.`



Certaines fonctions sont ambiguës en ce qui concerne les unités, et elles ne peuvent pas être utilisées avec des signaux auxquels sont associées des unités. Par exemple, la fonction Incrément est ambiguë en ce qui concerne les unités. Si vous utilisez des unités de distance, la fonction Incrément ne peut pas savoir s’il faut ajouter un mètre, un kilomètre ou un pied. En raison de cette ambiguïté, la fonction Incrément et les fonctions similaires ne peuvent pas être utilisées avec des données auxquelles sont associées des unités.

Une façon de contourner cette difficulté consiste à utiliser une constante numérique dont l’unité est correcte et la fonction d’addition sur le diagramme pour créer votre propre fonction “Ajouter une unité”.



Remarque

Vous ne pouvez pas utiliser d’unités dans les boîtes de calcul.

Unités polymorphes

Si vous voulez créer un VI qui calcule la moyenne quadratique d’une forme d’onde, vous devez définir l’unité associée à la forme d’onde. Un VI séparé est nécessaire pour les formes d’onde de tension, les formes d’onde de courant, les formes d’onde de température, et ainsi de suite. Pour qu’un VI effectue le même calcul, indépendamment des unités reçues par les entrées, le langage G offre la possibilité de travailler avec des unités polymorphes.

Créez une unité polymorphe en entrant \$x, où x est un nombre (par exemple, \$1). Considérez ceci comme étant un espace que vous réservez pour la véritable unité. Lorsque le VI est appelé, le G remplace toutes les occurrences de \$x dans ce VI par les unités que vous lui transmettez. Le symbole \$x est toujours remplacé par une unité exprimée en unités de base, par exemple, des mètres ou des mètres/sec, mais pas des pieds ni des pouces.

Une unité polymorphe est traitée comme une unité unique. Elle n’est convertible en aucune autre unité et elle se propage dans le diagramme exactement comme le font les autres unités. Lorsqu’une autre unité est connectée à un indicateur qui porte aussi l’abréviation \$1, une correspondance s’établit entre les unités et le VI peut être compilé.

§1 peut être utilisé dans des combinaisons comme n'importe quelle autre unité. Par exemple, si l'entrée est multipliée par 3 secondes avant d'être transmise à un indicateur, l'indicateur doit être en unités §1 s. Si l'indicateur a des unités différentes, le diagramme indique un mauvais fil de liaison.

Si vous devez utiliser plusieurs unités polymorphes, vous pouvez utiliser les abréviations §2, §3, et ainsi de suite.

Un appel à un sous-VI contenant des unités polymorphes calcule des unités de sortie basées sur les unités reçues par ses entrées. Par exemple, supposez que vous créez un VI avec deux entrées exprimées en unités polymorphes §1 et §2 qui crée une sortie de la forme §1 §2/s. Si le sous-VI reçoit des données en m/s pour l'entrée §1 et des données en kg pour l'entrée §2, l'unité de sortie est calculée comme étant des kg m/s².

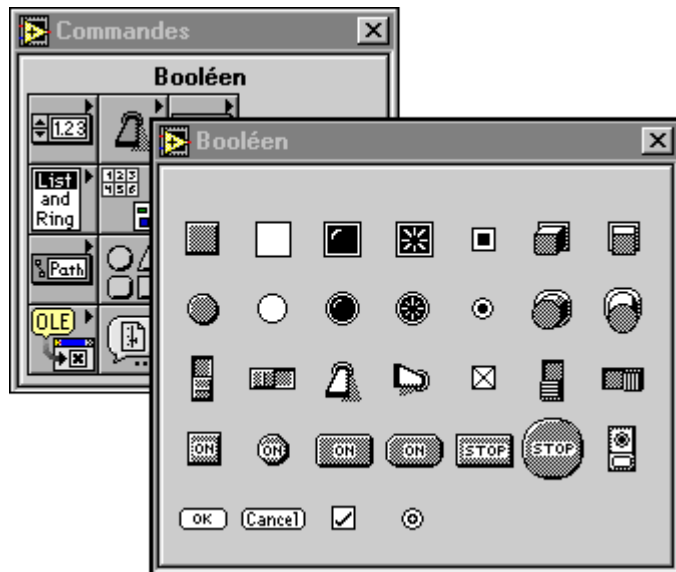
Supposez qu'un VI différent ait deux entrées en §1 et §1/s, et qu'il calcule une sortie en §1². Si ce VI reçoit des données en m/s pour l'entrée §1 et en m/s² pour l'entrée §1/s, l'unité de sortie est calculée comme étant des m²/s². Cependant, si ce VI reçoit des données en m pour l'entrée §1 et en kg pour l'entrée §1/s, l'appel du sous-VI est brisé. Une des entrées est déclarée en conflit d'unité et la sortie est calculée (si possible) à partir de l'autre. Un VI polymorphe peut avoir un sous-VI polymorphe parce que les unités respectives sont gardées distinctes.

Commandes et indicateurs booléens

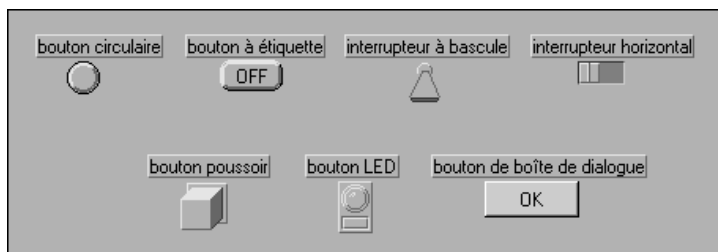
Ce chapitre décrit comment créer, utiliser et configurer les commandes et indicateurs booléens.

Création et utilisation de commandes et indicateurs booléens

Les commandes et indicateurs booléens ont deux valeurs : TRUE (Vrai) et FALSE (Faux). Les commandes et indicateurs booléens sont disponibles à partir de la palette **Commandes»Booléen**, comme le montre l'illustration suivante.



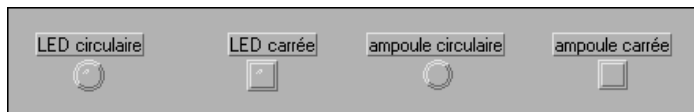
Certaines commandes booléennes simulant des boutons-poussoirs mécaniques, des interrupteurs à bascule et des interrupteurs à glissière sont représentés dans l'illustration suivante.



Remarque

Le bouton de boîte de dialogue, la case à cocher de boîte de dialogue et le bouton radio de boîte de dialogue ont une apparence différente sur chaque plate-forme. Pour plus d'informations sur les commandes de dialogue sur diverses plates-formes, consultez la section Personnalisation des commandes de boîte de dialogue du chapitre 8, Introduction aux objets de la face-avant.

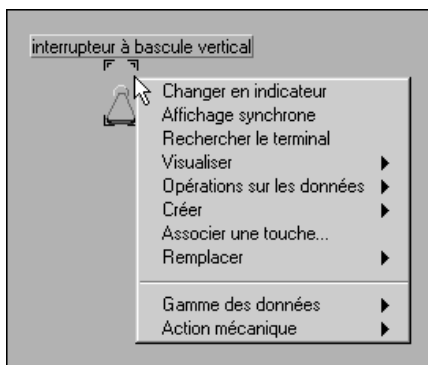
Certains indicateurs booléens simulant des LED et des voyants sont représentés dans l'illustration suivante.



En cliquant sur une commande booléenne avec l'outil Doigt, vous la faites passer alternativement de son état TRUE (marche) à son état FALSE (arrêt). En mode Exécution, cliquer sur un indicateur ne produit aucun effet, parce que les indicateurs sont destinés uniquement à une fonction de sortie. En mode Édition, vous pouvez agir à la fois sur les commandes et les indicateurs.

Configuration des commandes et indicateurs booléens

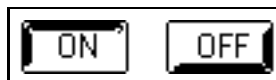
Chaque commande ou indicateur booléen a plusieurs éléments disponibles dans son menu local. Le menu local pour un objet booléen est représenté dans l'illustration ci-après.



Les éléments situés au-dessus de la ligne dans le menu local sont communs à toutes les commandes et à tous les indicateurs, et ils sont décrits dans la section *Options des commandes et indicateurs de la face-avant* du chapitre 8, *Introduction aux objets de la face-avant*. Les éléments **Gamme des données** et **Action mécanique** sont décrits plus loin dans ce chapitre.

Changer les étiquettes des booléens

Plusieurs commandes de la palette **Booléen** affichent du texte et sont appelées *Booléens à étiquette*. Initialement, les boutons affichent le mot ON dans leur état TRUE et le mot OFF dans leur état FALSE, comme le montre l'illustration suivante. Lorsque vous cliquez sur le bouton avec l'outil Doigt, la commande passe à l'état opposé. En mode Édition, utilisez l'outil Texte pour changer le texte dans n'importe lequel des deux états, par exemple, OUI au lieu de ON.



Par défaut, le texte est centré sur le bouton. Si vous voulez déplacer le texte, sélectionnez **Libérer le texte** dans le menu local.

Utilisez l'outil Flèche pour repositionner le texte, ou choisissez l'élément **Verrouiller le texte au centre**. Sélectionnez le texte booléen, puis utilisez le menu Police de caractères sur la barre d'outils pour changer la fonte, la

taille et la couleur du texte. Vous pouvez également cacher le texte booléen des deux états en désélectionnant l'élément **Texte booléen** du sous-menu **Visualiser**.

Les commandes booléennes qui n'apparaissent pas dans la palette en tant que booléens à étiquette sont considérés par défaut comme non étiquetés. Cependant, vous pouvez sélectionner **Texte booléen** à partir du sous-menu **Visualiser** pour leur attribuer une étiquette. Vous pouvez déplacer le texte booléen dans ces booléens. Leur texte n'est pas verrouillé au centre du bouton, à moins que vous ne sélectionniez **Verrouiller le texte au centre** à partir du menu local.

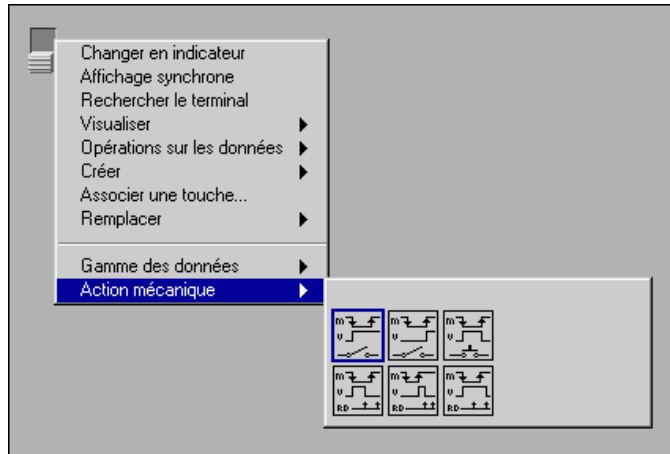
Si vous voulez changer la fonte de l'étiquette du nom ou du texte booléen d'un seul des deux états, sélectionnez l'étiquette de l'état que vous voulez changer avec l'outil Texte, puis utilisez les éléments du menu Police de caractères pour apporter les changements que vous désirez.

Contrôle de la gamme de données des booléens

Il peut être utile de détecter des erreurs parmi les valeurs booléennes. Si vous vous attendez à ce qu'un booléen soit toujours Vrai (TRUE), sélectionnez **Gamme des données»Interrompre si False** à partir du sous-menu du menu local du booléen. Si vous vous attendez à ce qu'un booléen soit toujours Faux (FALSE), sélectionnez **Gamme des données»Interrompre si True** pour détecter les erreurs. Si la valeur inattendue du booléen se produit, le VI s'interrompt avant ou après son exécution, comme cela est spécifié dans l'élément **Interrompre** de la gamme numérique.

Configuration de l'action mécanique des commandes booléennes

Les commandes booléennes ont six types d'action mécanique. Sélectionnez l'action appropriée pour votre application à partir de la palette **Action mécanique** du menu local, représentée dans l'illustration suivante. Parmi les symboles de cette palette, M désigne le mouvement du bouton de la souris lorsque vous utilisez la commande, V désigne la valeur de sortie de la commande, et RD désigne le moment auquel le VI lit la commande.



L'action **Commutation à l'appui** change la valeur de la commande à chaque fois que vous cliquez dessus avec l'outil Doigt, de façon semblable à un interrupteur de courant. L'action n'est pas affectée par la fréquence à laquelle le VI lit la commande.



L'action **Commutation au relâchement** change la valeur de la commande uniquement lorsque vous relâchez le bouton de la souris après avoir cliqué à l'intérieur des limites graphiques de la commande. L'action n'est pas affectée par la fréquence à laquelle le VI lit la commande.



L'action **Commutation jusqu'au relâchement** change la valeur de la commande uniquement lorsque vous cliquez dessus, et conserve la nouvelle valeur jusqu'à ce que vous relâchiez le bouton de la souris. Ensuite, la commande reprend sa valeur d'origine, imitant ainsi le principe de fonctionnement d'une sonnerie de porte. L'action n'est pas affectée par la fréquence à laquelle le VI lit la commande



Avec l'action **Armement à l'appui**, la commande change de valeur lorsque vous cliquez dessus, et elle conserve la nouvelle valeur jusqu'à ce que le VI la lise une fois. Ensuite, la commande reprend sa valeur par défaut, que vous continuiez ou non à appuyer sur le bouton de la souris. Cette action est similaire à celle d'un disjoncteur, et elle est utile pour arrêter des boucles While ou pour indiquer au VI d'effectuer une opération une fois seulement à chaque fois que vous activez la commande.



L'action **Armement au relâchement** change la valeur de la commande uniquement une fois que vous relâchez le bouton de la souris à l'intérieur des limites graphiques de la commande. Lorsque votre VI l'a lue une fois, la commande reprend son ancienne valeur. Cette action fonctionne de la même manière que les boutons de boîte de dialogue et les boutons du système.



L'action **Armement jusqu'au relâchement** change la valeur de la commande lorsque vous cliquez dessus et la conserve jusqu'à ce que votre VI la lise une fois ou que vous relâchiez le bouton de la souris, suivant ce qui se produit en dernier.

Pour un exemple de commandes et indicateurs booléens, consulter `exemples\general\controls\booleans.llb`.

Personnalisation des booléens avec des images importées

Vous pouvez créer votre propre style de booléen en important des images pour les états TRUE et FALSE de n'importe quels commandes ou indicateurs booléens. Ce processus est expliqué dans la section *Utilisation de l'éditeur de commandes* du chapitre 24, *Commandes personnalisées et définitions de type*.

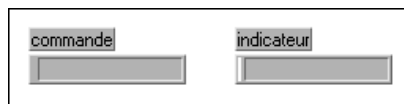
Commandes et indicateurs de chaîne de caractères

Ce chapitre décrit comment utiliser les commandes et indicateurs de chaîne de caractères, ainsi que la table. Vous pouvez accéder à ces objets au moyen de la palette **Commandes»Chaîne de caractères & table**, représentée dans l'illustration suivante.



Utilisation des commandes et indicateurs de chaîne de caractères

Une commande de chaîne de caractères et un indicateur de chaîne de caractères sont représentés dans l'illustration suivante.



Vous entrez ou changez du texte dans la commande de chaîne de caractères à l'aide de l'outil Doigt ou de l'outil Texte. Par défaut, le texte nouveau ou modifié n'est transféré vers le diagramme que lorsque vous appuyez sur la touche <Entrée> du pavé numérique, cliquez sur le bouton **Entrée** de la palette **Outils** ou cliquez en dehors de la commande pour terminer la session d'édition. Cependant, si vous sélectionnez l'élément **Mettre à jour la valeur en cours de frappe** à partir du menu local, la valeur de la commande change avec chaque caractère. Si vous appuyez sur la touche <Retour> du clavier alphanumérique, cela se traduit par un retour chariot.

Lorsque le texte atteint le bord droit de la zone d'affichage des chaînes de caractères, la chaîne de caractères se poursuit sur la ligne suivante, le saut de ligne s'effectuant sur un séparateur naturel tel qu'un espace ou un caractère de tabulation.



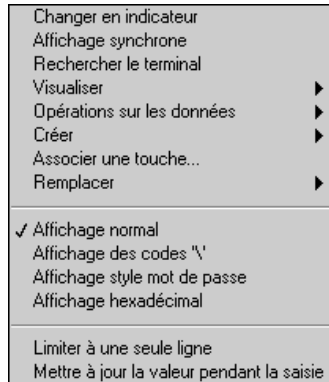
Remarque

*Lorsque vous exécutez un VI, vous utilisez la touche <Tab> pour passer à la commande suivante. Lorsque vous éditez un VI, vous appuyez sur la touche <Tab> pour changer d'outil. Pour entrer un caractère de tabulation dans une chaîne de caractères, sélectionnez l'élément de menu **Affichage des codes '\'** à partir du menu local de chaîne de caractères et tapez \t. Pour entrer un retour à la ligne dans une chaîne de caractères, appuyez sur la touche <Entrée> (Windows et HP-UX) ou <Retour> (Macintosh et Sun) sur le clavier alphanumérique, ou sélectionnez l'élément de menu **Affichage des codes '\'** à partir du menu local de chaîne de caractères et tapez \n. Pour entrer un retour chariot dans une chaîne de caractères, sélectionnez l'élément de menu **Affichage des codes '\'** à partir du menu local de chaîne de caractères et tapez \r. Consultez le tableau 11-1, Codes '\ du G dans la section **Types d'affichage de ce chapitre pour une liste complète des codes '\'**.*

Pour plus d'informations sur la manipulation des chaînes de caractères, consultez la rubrique **Référence en ligne»Function and VI Reference»Functions** des chaînes de caractères. De plus, reportez-vous aux exemples figurant dans `examples\general\strings.llb`.

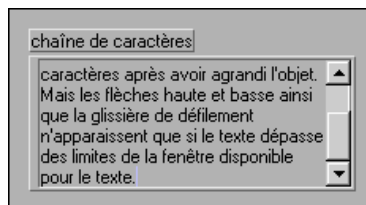
Éléments de menu des commandes et indicateurs de chaîne de caractères

Les chaînes de caractères ont des fonctions spéciales auxquelles vous pouvez accéder par l'intermédiaire du menu local, représenté dans l'illustration suivante.



Éléments de menu Barre de défilement

L'élément de menu **Barre de défilement** du sous-menu local de chaîne de caractères **Visualiser** est désactivé, à moins que vous n'augmentiez suffisamment la taille de votre chaîne de caractères pour y loger une barre de défilement. Si vous sélectionnez cet élément, une barre de défilement verticale apparaît sur la commande ou sur l'indicateur de chaîne de caractères, comme cela est représenté dans l'illustration suivante, afin que vous puissiez afficher le texte qui n'est pas visible dans la commande de chaîne de caractères. Vous pouvez aussi utiliser cet élément pour minimiser l'espace occupé sur la face-avant par les commandes de chaîne de caractères contenant une grande quantité de texte. Si l'élément de menu est grisé, vous devez, avant de pouvoir sélectionner cet élément, augmenter la hauteur de la chaîne de caractères pour qu'elle puisse accueillir la barre de défilement.



Types d’affichage

Les éléments de menu de la section centrale du menu local de chaîne de caractères vous permettent de spécifier si la chaîne de caractères doit afficher les données de manière normale, afficher des codes à barre oblique inverse (“\”) à la place des caractères non imprimables, afficher en mode de mot de passe (en utilisant le symbole “*” à la place de chaque caractère) ou afficher des caractères hexadécimaux.

Affichage normal

L’élément de menu **Affichage normal** affiche tous les caractères tels qu’ils sont tapés (à l’exception des caractères créés avec des touches spéciales tels que les touches <Tab> ou <Echap>, qui ne sont pas affichables).

Affichage des codes à barre oblique inverse (‘\’)

En choisissant **Affichage des codes ‘\’** à partir du menu local de chaîne de caractères, vous indiquez au logiciel d’interpréter les caractères se trouvant immédiatement après une barre oblique inverse (\) comme un code pour des caractères non affichables. Le tableau suivant montre comment le G interprète ces codes.

Tableau 11-1. Codes ‘\’ du G

Code	Interprétation du G
\00 - \FF	Valeur hexadécimale d’un caractère 8 bits, doit être en lettres majuscules
\b	Retour arrière (BS ASCII, équivalent à \08)
\f	Saut de page (FF ASCII, équivalent à \0C)
\n	Retour à la ligne (LF ASCII, équivalent à \0A)
\r	Retour chariot (CR ASCII, équivalent à \0D)
\t	Tab (HT ASCII, équivalent à \09)
\s	Espace (équivalent à \20)
\\	Barre oblique inverse (ASCII \, équivalent à \5C)

Utilisez des lettres majuscules pour les caractères hexadécimaux et des lettres minuscules pour les caractères spéciaux, tels qu’un saut de page ou un retour arrière. Le G interprète par exemple la séquence `\bFare` comme BF hexadécimal suivi du mot *are*, alors que le G interprète `\bFare` et

`\bfare` comme un retour arrière suivi des mots *Fare* et *fare*. Dans la séquence `\Bfare`, `\B` n'est pas le code de retour arrière, et `\Bf` n'est pas un code hexadécimal valide. Dans un cas comme celui-ci, lorsqu'une barre oblique inverse est suivie seulement d'une partie d'un caractère hexadécimal valide, le G suppose qu'un θ suit la barre oblique inverse, donc le G interprète `\B` comme *OB hexadécimal*. À chaque fois qu'une barre oblique inverse n'est pas suivie d'un caractère hexadécimal valide, le G ignore le caractère de barre oblique inverse.

Vous pouvez entrer certains caractères non affichables à partir du clavier, tels qu'un retour chariot, dans une commande de chaîne de caractères, que vous ayez ou non sélectionné **Affichage des codes** '`\`'. Cependant, si vous activez le mode de barre oblique inverse lorsque la fenêtre d'affichage contient du texte, le G rafraîchit l'écran pour afficher la représentation en barre oblique inverse de tous les caractères non affichables, ainsi que le caractère `\` lui-même.

Supposez que le mode soit configuré sur Affichage normal et que vous entriez la chaîne de caractères suivante.

```
left
\righ\3F
```

Lorsque vous activez le mode Affichage des codes '`\`', la chaîne de caractères suivante apparaît, étant donné que le retour chariot après `left` et les barres obliques inverses qui suivent sont représentés sous forme de barre oblique inverse par `\n\`.

```
left\n\righ\3F
```

Supposez maintenant que vous sélectionniez **Affichage des codes** '`\`' et que vous entriez la chaîne de caractères suivante.

```
left
\righ\3F
```

Lorsque vous désactivez le mode Affichage des codes '`\`', la chaîne suivante apparaît, le G ayant interprété à l'origine `\r` comme un retour chariot et en affichant maintenant un. `\3F` est la représentation spéciale du point d'interrogation (?), et la chaîne est affichée de la manière suivante.

```
left
ight?
```

Maintenant, si vous sélectionnez à nouveau **Affichage des codes** ‘\’, la chaîne de caractères suivante apparaît.

```
left\n\righ?
```

Les indicateurs se comportent de la même façon.

Dans ces exemples, les données de la chaîne de caractères ne changent pas d’un mode à l’autre. C’est seulement la représentation affichée de certains caractères qui change.

Le mode de barre oblique inverse est utile pour mettre au point des programmes et pour envoyer des caractères non imprimables à des instruments, des ports série et d’autres périphériques.

Affichage style mot de passe

Avec l’élément **Affichage style mot de passe**, la commande de chaîne de caractères affiche le symbole “*” pour chaque caractère qui est entré dans celle-ci. Toutefois, lorsque vous lisez les données de la chaîne de caractères à partir du diagramme, vous lisez les données que l’utilisateur a effectivement entrées. Si vous essayez de copier des données à partir de la commande, seuls les caractères * sont copiés.

Affichage hexadécimal

Utilisez l’élément **Affichage hexadécimal** pour afficher la chaîne de caractères sous forme de caractères hexadécimaux plutôt que de caractères alphanumériques. Tout comme l’**Affichage des codes** ‘\’, l’élément **Affichage hexadécimal** est utile pour mettre au point des programmes et pour communiquer avec des instruments.

Limiter à une seule ligne

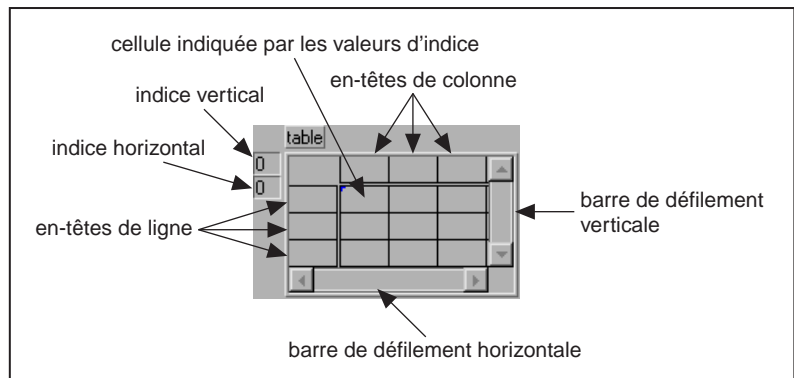
L’élément de menu **Limiter à une seule ligne** vous empêche d’entrer un retour chariot dans une chaîne de caractères pendant que vous la tapez.

Mettre à jour la valeur en cours de frappe

Sélectionnez l'élément de menu Mettre à jour la valeur en cours de frappe pour permettre que la valeur d'une commande change à mesure que les caractères sont entrés au lieu d'attendre que l'utilisateur clique sur le bouton Entrée ou termine d'une autre manière la session d'édition. Ce mode peut être utile pour vérifier l'exactitude de l'entrée, limiter l'entrée ou donner un retour d'information à l'utilisateur. Par exemple, il est possible de créer une commande de chaîne de caractères qui limite l'entrée à des caractères alphanumériques.

Tables

Une table est un tableau 2D de chaînes de caractères. L'illustration suivante représente un exemple de table sur lequel sont représentées toutes ses fonctions.

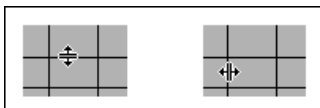


Une table a des en-têtes de rangées et de colonnes qui sont séparés des données par un mince espace à bords ouverts. Vous pouvez entrer des en-têtes lorsque vous placez la table sur la face-avant, et vous pouvez les changer à l'aide de l'outil Doigt ou de l'outil Texte. Vous pouvez mettre à jour ou lire les en-têtes à l'aide d'un attribut node.

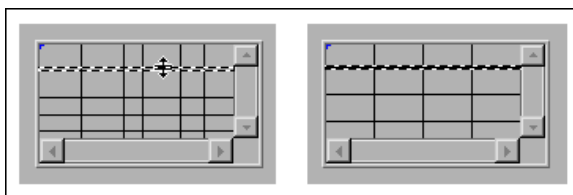
L'affichage des indices indique quelle cellule est visible dans le coin supérieur gauche de la table. Vous pouvez manipuler ces indices exactement de la même manière que pour un tableau.

Redimensionnement des tables, des rangées et des colonnes

Vous pouvez redimensionner la table à partir de n'importe quel coin avec l'outil Coin. Vous pouvez redimensionner des rangées et colonnes séparément dans une table en déplaçant les lignes de séparation avec l'outil Flèche. Lorsque l'outil est positionné correctement pour déplacer une ligne, un des curseurs de déplacement représentés dans l'illustration suivante apparaît. Cliquez sur une ligne et déplacez-la pour redimensionner la rangée ou la colonne.



Vous pouvez utiliser la touche <Maj> pendant que vous déplacez une ligne de séparation pour dimensionner plusieurs rangées ou colonnes de sorte qu'elles soient toutes de la même taille. Si la rangée ou la colonne que vous redimensionnez se trouve à l'intérieur d'une zone de la table que vous avez sélectionnée (délimitée en bleu ou en caractères gras), toutes les rangées ou colonnes de la table sont également ramenées à la même dimension. Par exemple, les rangées horizontales irrégulières de l'illustration de gauche ci-dessous deviennent régulièrement espacées dans l'illustration de droite parce que la touche <Maj> a été utilisée.



Entrer et sélectionner des tables de données

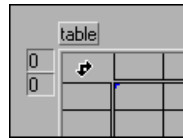
Vous pouvez utiliser le clavier pour entrer rapidement des données dans une table. Cliquez à l'intérieur d'une cellule avec l'outil Doigt ou avec l'outil Texte, puis tapez vos données.

La touche <Entrée> du clavier alphanumérique entre votre texte et déplace le curseur dans la cellule qui se trouve en-dessous. La touche <Entrée> du pavé numérique entre vos données et valide la saisie. Si vous appuyez sur la touche <Maj> en même temps que sur les touches fléchées, le curseur d'entrée passe dans les cellules adjacentes.

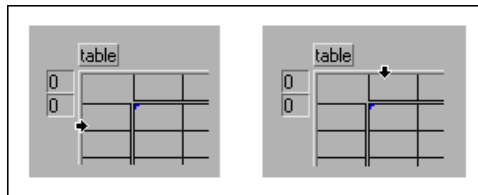
Vous pouvez sélectionner des cellules séparément avec l’outil Doigt ou en double-cliquant dessus, et vous pouvez étendre la sélection en appuyant sur <Maj-clic> et en faisant glisser le curseur.

Le fait de déplacer le curseur en dehors du contenu actuel de la table fait défiler la table tout en étendant la sélection. Une bordure apparaît autour des cellules sélectionnées pour indiquer qu’elles sont sélectionnées. Vous pouvez activer ou désactiver le défilement à l’aide de l’élément **Défilement de la sélection** du menu local de la table.

Vous pouvez également sélectionner toutes les données d’une table, ou toutes les données d’une rangée ou d’une colonne. Pour cela, positionnez l’outil Doigt dans le coin supérieur gauche de la table pour sélectionner toutes les données. Un curseur spécial, en forme de double flèche, apparaît lorsque l’outil est positionné correctement, comme le montre l’illustration suivante. Cliquez dans cette zone pour sélectionner les données.



Pour sélectionner une rangée ou une colonne entière de données, positionnez l’outil Doigt sur le bord gauche d’une rangée, ou en haut d’une colonne. A nouveau, un curseur spécial en forme de flèche apparaît lorsque l’outil est correctement positionné, comme le montre l’illustration suivante. Cliquez et faites glisser le curseur sur le nombre de colonnes ou sur le nombre de rangées désiré pour sélectionner les données.



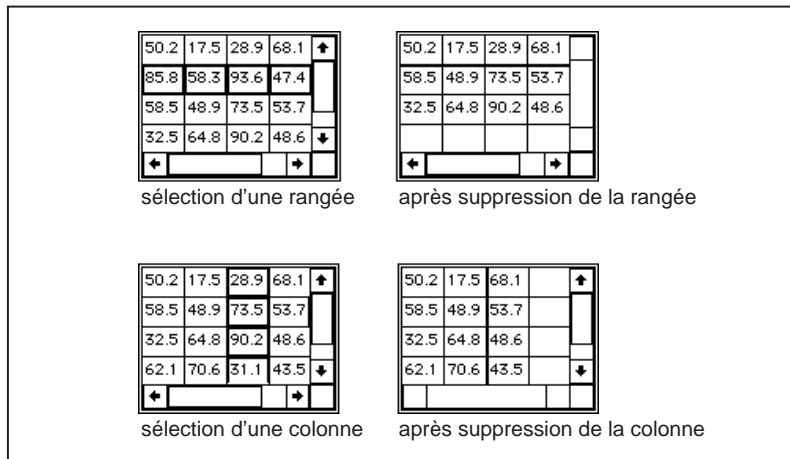
Vous pouvez copier, couper et coller des données à l’aide des éléments de menu **Copier des données**, **Couper des données** et **Coller des données** qui se trouvent dans le sous-menu **Opérations sur les données** du menu local.

Si vous coupez une ou plusieurs rangées de données, toutes les rangées situées en dessous de celles-ci remontent, comme le montre la partie supérieure de l’illustration suivante. Si vous coupez une colonne de

données, toutes les colonnes situées à sa droite se déplacent vers la gauche, comme le montre la partie inférieure de l'illustration suivante.

Si vous coupez une partie d'une rangée ou d'une colonne, la rangée ou la colonne est supprimée en entier de la table, mais seules les données sélectionnées sont copiées dans le Presse-papiers.

Si vous collez sans avoir d'abord sélectionné une zone cible, des rangées et des colonnes supplémentaires sont ajoutées à la table lorsque cela est nécessaire.



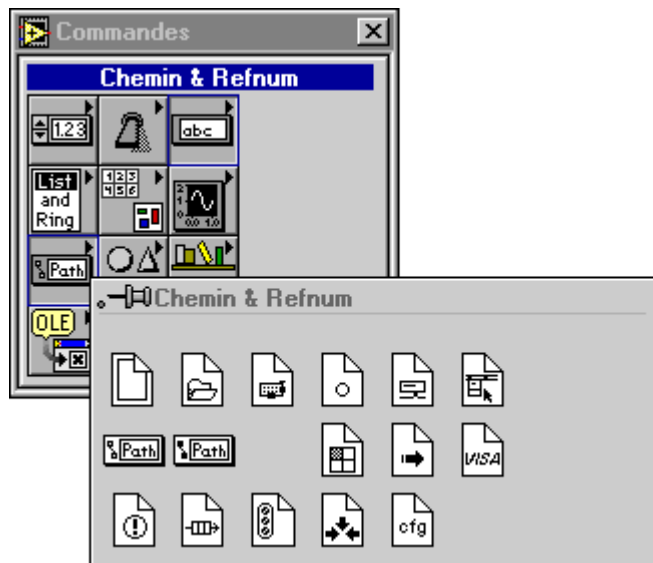
Pour visualiser une zone de données sélectionnée, vous pouvez activer ou désactiver l'élément en sélectionnant ou en désélectionnant **Opérations sur les données»Visualiser la sélection**.

Les en-têtes des rangées et des colonnes d'une table ne font pas partie des données de la table. Les en-têtes de la table constituent un ensemble de données séparé, et vous pouvez les lire et les définir à l'aide d'un attribut node.

En ce qui concerne le diagramme, une table est équivalente à un tableau à deux dimensions de chaînes de caractères. C'est pourquoi les fonctions de chaînes de caractères peuvent être utilisées pour manipuler les tables. Pour plus d'informations sur l'utilisation de ces fonctions, consultez le topic **Référence en ligne»Function and VI Reference»String Functions**.

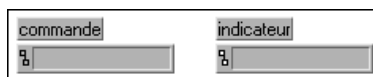
Commandes et indicateurs de chemin et refnum

Ce chapitre décrit comment utiliser les commandes de chemin de fichier et les refnums, qui sont disponibles à partir de la palette **Commandes» Chemin & Refnum**, représentée dans l'illustration suivante.



Commandes et indicateurs de chemin

Une commande de chemin et un indicateur de chemin sont représentés dans l'illustration suivante.



Vous utilisez des commandes et indicateurs de chemin pour entrer et afficher, respectivement, l'emplacement d'un fichier ou d'un répertoire



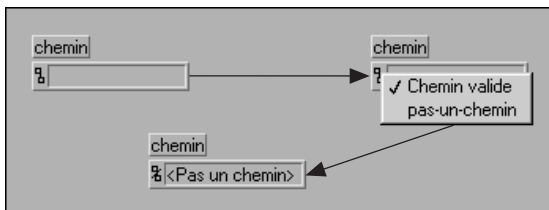
Symbole Chemin



Symbole Pas un chemin

dans un système de fichiers, à l'aide de la syntaxe standard pour une plate-forme donnée. Si une fonction qui est supposée renvoyer un chemin échoue, elle renvoie un chemin invalide. Lorsqu'un chemin invalide est affiché par une commande ou un indicateur de chemin, le symbole Chemin prend l'apparence du symbole Pas un chemin, et <Pas un chemin> apparaît dans le champ de texte pour indiquer que le chemin est invalide.

Vous pouvez changer la valeur d'une commande de chemin valide en chemin invalide en cliquant sur le symbole Chemin et en sélectionnant l'élément **Pas un chemin** dans le menu. Celui-ci est représenté dans l'illustration suivante. De la même manière, vous pouvez changer la valeur d'une commande ou d'un indicateur de chemin invalide en chemin valide en cliquant sur le symbole Pas un chemin et en sélectionnant l'élément **Chemin valide** du menu.



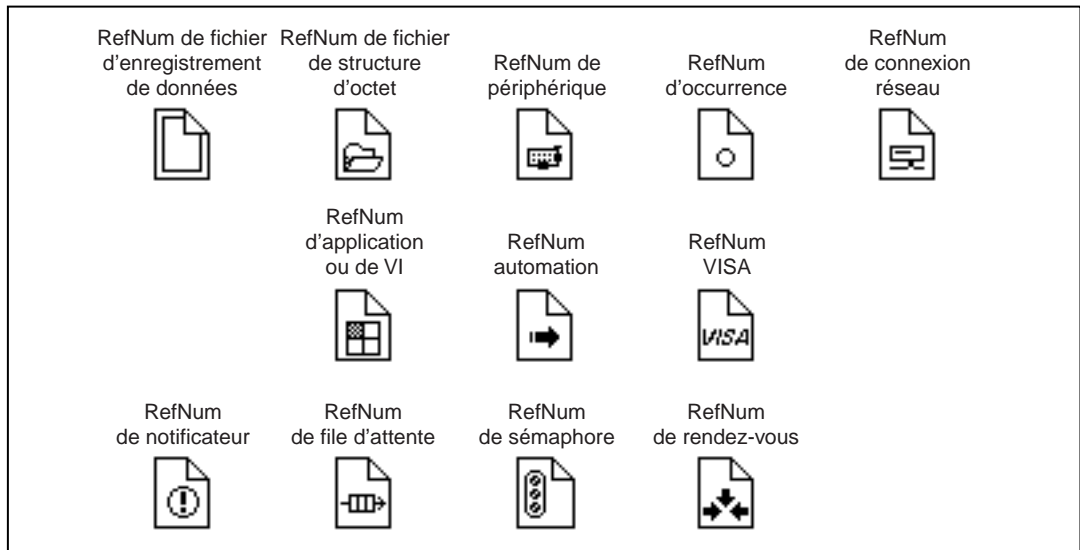
Vous pouvez utiliser la valeur **Pas un chemin** comme valeur par défaut pour une commande de chemin de votre VI. De cette manière, vous pouvez détecter si l'utilisateur n'a pas entré de chemin. Dans ce cas, utilisez une boîte de dialogue de fichier pour demander un chemin.

Un chemin vide apparaît dans une commande de chemin sous la forme d'une chaîne de caractères vide. Lorsqu'il est envoyé aux fonctions E/S de fichiers, en particulier à "Info sur le fichier/répertoire" et à "Lister les répertoires", ce chemin vide fait référence, sur la plate-forme Windows, à la liste des lecteurs reliés à votre ordinateur. Sur la plate-forme Mac, le chemin vide fait référence aux fichiers qui se trouvent sur votre bureau. Sur les plates-formes UNIX, le chemin vide fait référence au répertoire racine.

Commandes et indicateurs de refnum

La palette **Chemin & Refnum** contient plusieurs commandes de refnum, représentées dans l'illustration suivante. Un refnum est un identificateur unique utilisé pour identifier un objet, tel qu'un fichier, un périphérique, ou une connexion réseau avec une autre machine. De nombreux refnums sont utilisés pour spécifier l'objet sur lequel doit agir une opération d'E/S, comme la lecture de fichier par la fonction "Lire un fichier". Une commande de refnum transmet un refnum à un VI et un indicateur de refnum transmet un refnum provenant d'un VI. Pour chaque commande de refnum, il existe un indicateur correspondant. Vous construisez un indicateur de refnum en créant d'abord une commande de refnum. Ensuite, ouvrez le menu local et sélectionnez **Changer en indicateur** dans le menu.

L'illustration suivante représente les différentes sortes de refnums.



Lorsque vous ouvrez un fichier, vous devez spécifier le chemin du fichier que vous voulez ouvrir. Un *refnum* identifiant ce fichier est alors renvoyé. Ce refnum est utilisé dans toutes les opérations ultérieures ayant trait à ce fichier. Vous pouvez considérer ce refnum comme un numéro unique produit à chaque fois que vous ouvrez un fichier et destiné à l'identifier. Lorsque vous fermez le fichier, le refnum est dissocié du fichier. Ci-après se trouvent les descriptions des refnums représentés dans l'illustration ci-dessus, et qui se trouvent dans la palette **Commandes» Chemin & Refnum**.

Il existe deux types de refnums de fichiers, un pour les fichiers d'enregistrement de données et un autre pour les fichiers de structure d'octet.

- **RefNum de fichier d'enregistrement de données** : étant donné que les fichiers d'enregistrement de données ont une structure inhérente, le RefNum de fichier d'enregistrement de données transmet aux VIs appelants (ou reçoit de leur part) le refnum, ainsi qu'une description du type de fichier. Le RefNum de fichier d'enregistrement de données peut être redimensionné, comme un cluster. Vous placez une commande à l'intérieur du refnum définissant la structure du fichier. Pour un fichier contenant des nombres, vous devez créer un refnum de fichier d'enregistrement de données contenant un nombre. Si chaque enregistrement du fichier contient deux nombres, vous devez placer un cluster à l'intérieur du refnum, et placer deux commandes numériques à l'intérieur du cluster.
- **RefNum de fichier de structure d'octet** : le RefNum de fichier de structure d'octet est utilisé avec des fichiers de structure d'octet, ceux-ci pouvant être aussi bien des fichiers texte que des fichiers binaires. Typiquement, vous devez l'utiliser lorsque vous ouvrez ou créez un fichier dans un VI mais que vous voulez effectuer une E/S sur ce même fichier dans un autre VI. Vous devez avoir une commande de refnum sur la face-avant du VI pour effectuer l'E/S et un indicateur de refnum sur la face-avant du VI pour ouvrir ou créer le fichier.

Les autres refnums sont faciles à utiliser, parce qu'ils n'ont aucune option.

- **RefNum de périphérique** : le RefNum de périphérique est utilisé avec les fonctions d'E/S de périphérique disponibles sur Macintosh. Il est utilisé uniquement pour ouvrir un périphérique dans un VI et pour effectuer une E/S sur le même périphérique dans un autre VI.



Remarque

Une commande de RefNum de périphérique est rarement nécessaire sur votre face-avant. Ces commandes sont utilisées dans les VIs E/S d'instrument de vi.lib pour accéder aux drivers de périphériques série en G. Une commande de RefNum de périphérique est uniquement nécessaire pour écrire un driver de périphérique en G à fonction spécifique pour le Macintosh.

- **RefNum d'occurrence** : le RefNum d'occurrence est utilisé avec les fonctions d'occurrence. Vous ne devez l'utiliser que lorsque vous voulez générer une occurrence dans un VI et définir ou attendre l'occurrence dans un autre VI.

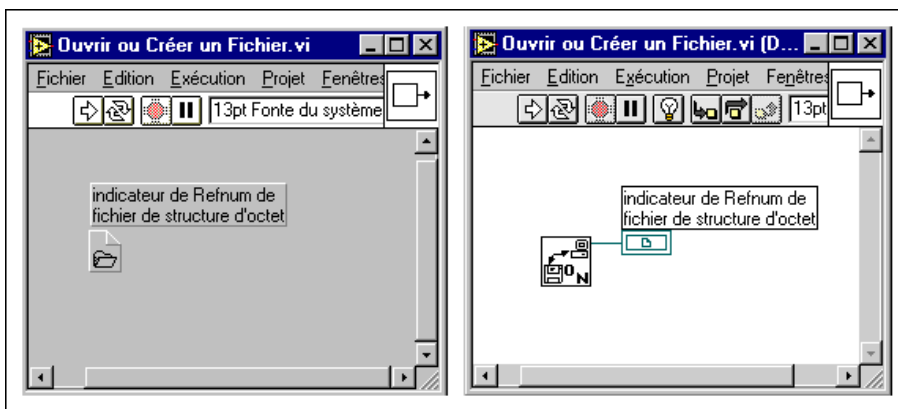
- **RefNum de connexion réseau** : le RefNum de connexion réseau est utilisé avec les VIs TCP/IP. Vous devez l'utiliser généralement lorsque vous voulez ouvrir une connexion réseau dans un VI et effectuer une E/S sur la connexion réseau dans un autre VI.
- **Refnum de VI ou d'application** : la commande de refnum de VI ou d'application est utilisée avec les fonctions VI Serveur. Cette commande de refnum est utilisée lorsque vous ouvrez une référence à une application LabVIEW, ou à un VI au sein de LabVIEW, et que vous souhaitez transmettre la référence comme paramètre à un autre VI. En transmettant l'un de ces refnums à l'une des fonctions VI Serveur, vous pouvez contrôler le comportement de l'application et de certains VIs spécifiques. En sélectionnant **Sélectionner la classe de VI Serveur** à partir du sous-menu de commande de refnum, vous pouvez spécifier que le type de données de la commande soit un refnum d'application, un refnum de VI ou un refnum de VI de type strict. Les refnums de VI de type strict comportent des informations sur le type de données qui comprennent le cadre connecteur du VI et peuvent par conséquent être utilisées pour appeler un VI chargé dynamiquement à l'aide de la fonction "Appel par référence". Vous pouvez sélectionner dans le disque un VI qui a le cadre connecteur désiré en utilisant **Sélectionner une classe de VI Serveur»**
Parcourir... Vous pouvez également faire glisser et déposer un cadre connecteur ou une icône de VI, ou encore une icône de sous-VI, d'un diagramme ou de la fenêtre de hiérarchie jusqu'à une commande de refnum de VI pour spécifier le type d'une commande de refnum de VI de type strict.

Alors que les commandes de refnum de VI sont généralement utilisées pour transmettre un refnum de VI d'un VI à un autre, une commande de refnum de VI de type strict doit également être fournie en entrée, en tant que spécificateur de type, à la fonction "Ouvrir une référence de VI" lorsque vous voulez obtenir une référence de type strict à un VI. Dans ce cas, la valeur de la commande de refnum est sans importance. Seul le type est utilisé par la fonction.

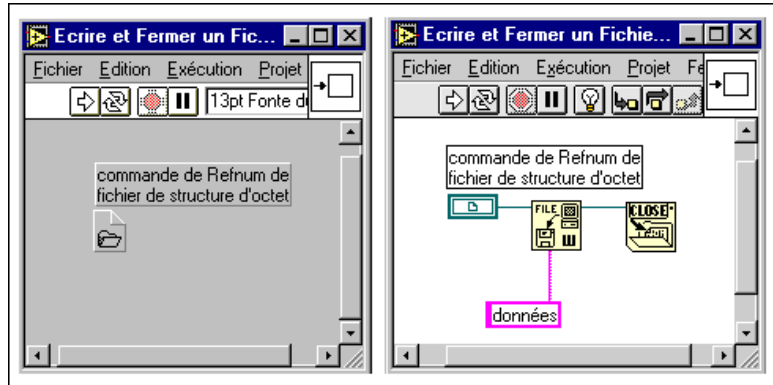
- **Refnum Automation** : le refnum Automation est une référence à un objet Automation.
- **RefNum VISA** : Un RefNum VISA est une référence à une ressource VISA.
- **RefNum de notificateur** : le RefNum de notificateur est utilisé avec les VIs de notification. Vous devez l'utiliser généralement lorsque vous voulez créer un notificateur dans un VI et attendre une notification ou envoyer une notification dans un autre VI.

- **RefNum de file d'attente** : le RefNum de file d'attente est utilisé avec les VIs de file d'attente. Vous devez l'utiliser généralement lorsque vous voulez créer une file d'attente dans un VI et insérer ou retirer des éléments de la file d'attente dans un autre VI.
- **RefNum de sémaphore** : le RefNum de sémaphore est utilisé avec les VIs de sémaphore. Vous devez l'utiliser généralement lorsque vous voulez créer un sémaphore dans un VI et acquérir ou relâcher le sémaphore dans un autre VI.
- **RefNum de rendez-vous** : le RefNum de rendez-vous est utilisé avec les VIs de rendez-vous. Vous devez l'utiliser généralement lorsque vous voulez créer un rendez-vous dans un VI et attendre le rendez-vous dans un autre VI.

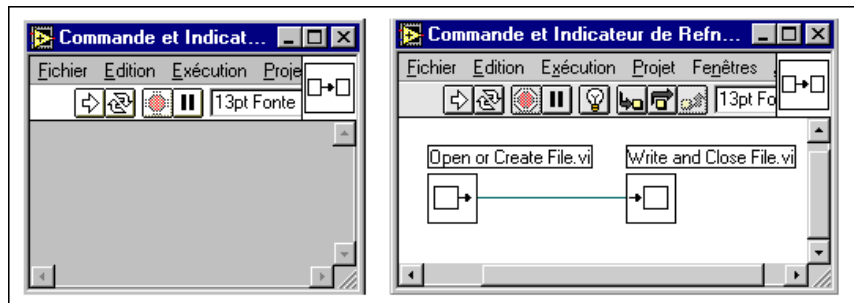
L'illustration suivante représente la face-avant et le diagramme d'un VI qui ouvre ou crée un fichier et transmet un refnum de fichier de structure d'octet, par l'intermédiaire d'un indicateur de refnum de fichier de structure d'octet, qui peut être utilisé pour accéder au fichier dans d'autres VIs.



L'illustration suivante représente la face-avant et le diagramme d'un VI qui écrit des données dans le fichier spécifié par le refnum de fichier de structure d'octet transmis au VI par l'intermédiaire d'une commande de refnum de fichier de structure d'octet, puis qui ferme ce fichier.



L'illustration suivante représente la face-avant et le diagramme d'un VI qui transmet un refnum de fichier de structure d'octet utilisé pour accéder à un fichier ouvert ou créé par un sous-VI à un autre sous-VI qui écrit des données dans ce fichier puis le ferme.

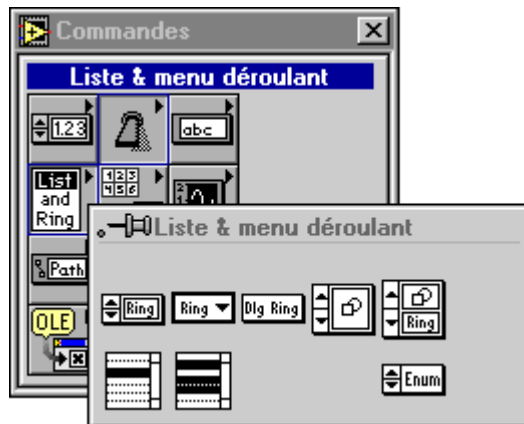


Remarque

Utilisez une commande ou un indicateur de refnum si vous voulez transmettre un refnum à un VI ou en provenance d'un VI.

Commandes et indicateurs de type liste et menu déroulants

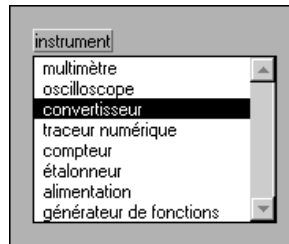
Ce chapitre décrit les commandes et indicateurs de type liste déroulante et de type menu déroulant, qui sont disponibles à partir de la palette **Commandes»Listes & menus déroulants** et qui sont représentés dans l'illustration suivante.



La palette **Listes & menus déroulants** contient huit commandes : Menu déroulant texte, Menu déroulant simple, Menu déroulant de boîte de dialogue, Menu déroulant d'images, Menu déroulant texte & images, Liste déroulante à sélection unique, Liste déroulante à sélection multiple et Type énuméré. Ce chapitre aborde les commandes de type menu déroulant, les commandes de type liste déroulante et les commandes de type énuméré, dans cet ordre. Pour plus d'informations sur les attributs des commandes de type liste déroulante et menu déroulant, consultez le chapitre 22, *Attribute Nodes*.

Commandes à liste déroulante

Les commandes de type liste déroulante offrent aux utilisateurs une liste d'éléments. Un exemple est représenté dans l'illustration suivante. Il existe deux types de commandes de type liste déroulante : A sélection unique, pour laquelle un seul élément peut être sélectionné, et A sélection multiple, pour laquelle un ou plusieurs éléments peuvent être sélectionnés.



Vous avez la possibilité d'associer des symboles aux éléments de texte (comme dans la boîte de dialogue **Enregistrer**, où les répertoires et les fichiers ont des symboles différents). Vous pouvez également désactiver des éléments individuels et placer des séparateurs de lignes entre les éléments. Vous détectez les éléments actuellement sélectionnés en lisant la valeur de la commande. En utilisant un attribut node, vous pouvez également détecter sur quel élément l'utilisateur a double-cliqué, le cas échéant. Enfin, vous pouvez définir le texte et le symbole des éléments et désactiver les éléments à l'aide d'un attribut node.

Création d'une liste d'éléments

Lorsque vous posez une liste déroulante sur une face-avant, elle ne contient aucun élément. Vous devez créer la liste d'éléments, en mode d'édition, en les tapant avec l'outil Texte, ou en mode d'exécution, en utilisant un attribut node. Séparez les éléments les uns des autres en entrant un retour chariot. Si vous pouvez déterminer les éléments lorsque vous créez le VI, vous pouvez les taper pendant que vous êtes en mode d'édition. Si vous ne pouvez les déterminer qu'au moment de l'exécution (par exemple, une liste de fichiers), utilisez un attribut node.

Sélection des éléments d'une liste déroulante

Il y a trois façons de sélectionner des éléments dans une liste déroulante : en utilisant la souris, en utilisant les touches fléchées ou en tapant une partie du nom de l'élément que vous souhaitez.

- **Souris** : utilisez l'outil Doigt pour sélectionner des éléments. Avec la Liste déroulante à sélection multiple, vous pouvez sélectionner plusieurs éléments en <Ma j>-cliquant sur d'autres éléments.
- **Touches fléchées** : utilisez les touches fléchées pour sélectionner des éléments. Pour ajouter l'élément couramment mis en évidence à la liste d'éléments sélectionnés, appuyez sur la barre d'espace. Vous pouvez désélectionner un élément couramment sélectionné en utilisant les touches fléchées pour atteindre l'élément en question, puis en appuyant sur la barre d'espace.
- **taper une partie du nom de l'élément que vous voulez** : tapez le nom d'un élément pour sélectionner l'élément en question. Vous pouvez également utiliser les touches fléchées gauche ou droite pour atteindre l'élément précédent ou l'élément suivant correspondant aux lettres tapées.

Types de données des listes déroulantes

Le type de données d'une Liste déroulante à sélection unique est Int32. La valeur d'une donnée de la Liste déroulante à sélection unique est un nombre qui représente l'élément couramment sélectionné, le premier élément ayant une valeur égale à zéro. Si aucun élément n'est sélectionné, la valeur est égale à -1.

La Liste déroulante à sélection multiple est un tableau de Int32, et les valeurs du tableau représentent les éléments couramment sélectionnés. Si aucun élément n'est sélectionné, la valeur est un tableau vide.

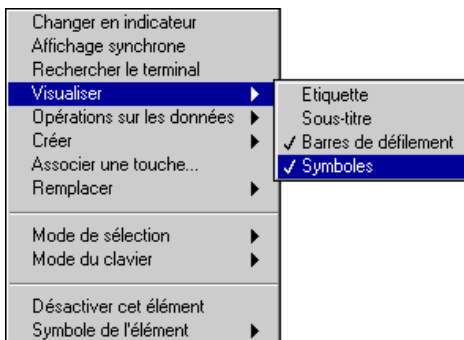
Éléments du menu local de liste déroulante

Le menu local de liste déroulante comprend les éléments représentés dans l'illustration suivante. Les éléments spécifiques aux commandes à liste déroulante sont décrits dans cette section.

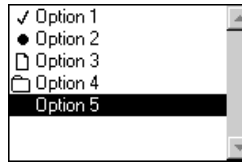


Visualiser

Vous utilisez l'élément **Visualiser** du menu local, représenté dans l'illustration suivante, pour sélectionner les composants de la liste déroulante qui doivent être visibles. Vous pouvez ainsi choisir de visualiser ou de masquer l'étiquette, la barre de défilement et les symboles.



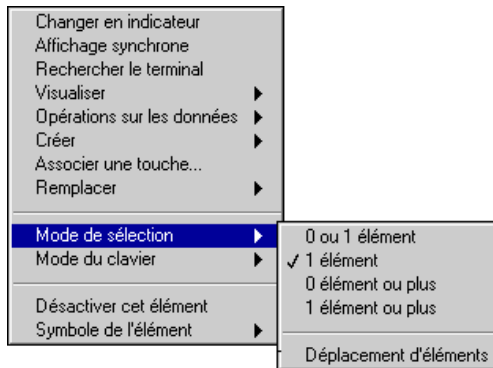
Si vous sélectionnez **Visualiser»Symboles**, une colonne supplémentaire apparaît dans la liste déroulante, sur le côté gauche de la liste, pour permettre l'affichage des symboles. Un exemple vous est présenté dans l'illustration suivante. Par défaut, aucun élément n'a de symbole. Vous pouvez ajouter des symboles à l'aide de **Symbole de l'élément** ou d'un **Attribute Node**.



Mode de sélection

Vous utilisez **Mode de sélection** à partir du menu local pour spécifier le nombre d'éléments que vous pouvez sélectionner à la fois.

Pour une Liste déroulante à sélection unique, les éléments du mode de sélection configurent la liste déroulante pour qu'elle autorise zéro ou une sélection, ou une sélection, comme cela est représenté dans l'illustration suivante.



Si vous sélectionnez **1 élément** (le paramètre par défaut), un élément de la liste déroulante est toujours sélectionné (mis en évidence). Lorsque vous cliquez sur un élément différent, le nouvel élément est sélectionné (mis en évidence), et l'élément précédent est désélectionné. Si vous sélectionnez **0 ou 1 élément**, le comportement est le même, à ceci près qu'un utilisateur peut désélectionner l'élément couramment sélectionné en <Ma j> -cliquant dessus, après quoi aucun élément n'est sélectionné. Vous pouvez utiliser **Déplacer l'élément** pour amener des éléments dans la liste déroulante en les faisant glisser avec la souris.

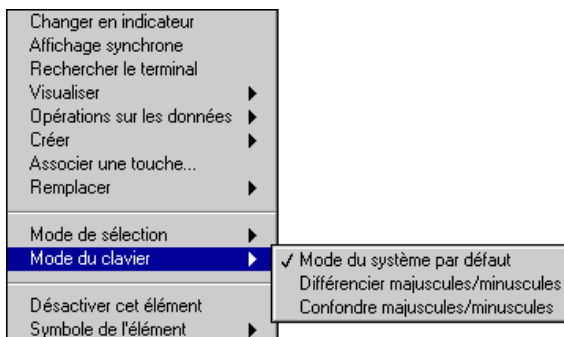
Dans une Liste déroulante à sélection multiple, vous avez les mêmes éléments de mode de sélection que dans la Liste déroulante à sélection unique. De plus, vous pouvez rendre possible la sélection de plusieurs éléments, comme cela est représenté dans l'illustration suivante.



Par défaut, la Liste déroulante à sélection multiple a un mode de sélection dans lequel l'utilisateur peut sélectionner **0 élément ou plus**. Vous pouvez également sélectionner **1 élément ou plus** pour le mettre dans un mode de sélection multiple où au moins un élément doit être sélectionné.

Mode du clavier

Vous pouvez utiliser le **Mode du clavier** à partir du menu local, représenté dans l'illustration suivante, pour indiquer comment la liste déroulante traite les caractères en lettres majuscules et minuscules lorsque vous sélectionnez des éléments en tapant leurs premières lettres.

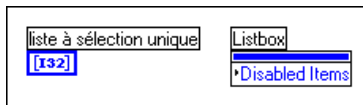


Si vous sélectionnez **Différencier majuscules/minuscules**, un élément est sélectionné uniquement si les lettres majuscules et minuscules tapées par l'utilisateur correspondent exactement à l'élément. Si vous sélectionnez **Confondre majuscules/minuscules**, la casse n'est pas prise en compte. Si vous sélectionnez **Mode du système par défaut** (le paramètre par défaut pour une liste déroulante), la liste déroulante se comporte de la même manière que les autres listes déroulantes pour une plate-forme donnée. Notamment, sous Windows et sur Macintosh, la liste déroulante ignore la casse. Sous UNIX, la liste déroulante respecte la casse.

Désactiver un élément

Vous pouvez désactiver un élément d'une liste déroulante en ouvrant le menu local de l'élément et en sélectionnant **Désactiver cet élément** à partir du menu local. Pour activer un élément désactivé, ouvrez le menu local de l'élément et sélectionnez **Activer cet élément**.

Pour activer/désactiver un élément à partir d'un programme, utilisez l'élément **Disabled Items** de l'Attribute Node, comme cela est représenté dans l'illustration suivante.

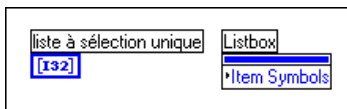


Symbole d'un élément et ligne de division

Vous pouvez définir le symbole d'une liste déroulante en ouvrant le menu local de l'élément et en sélectionnant l'un des éléments du sous-menu **Symbole de l'élément**. Si vous sélectionnez l'élément qui se trouve tout en bas à droite, vous remplacez l'élément de la liste par une ligne de division.



Pour définir des symboles d'éléments d'une liste à partir d'un programme, utilisez l'élément de **Item Symbols** l'Attribute Node (qui est un tableau de symboles d'éléments) représenté dans l'illustration suivante.

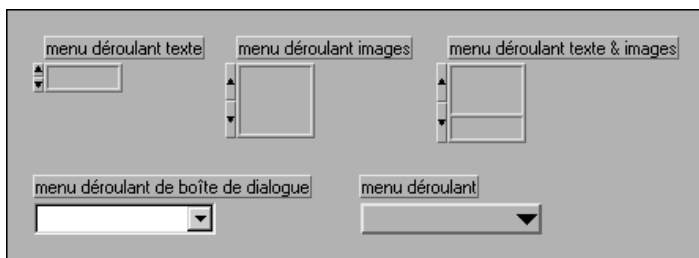


Vous pouvez spécifier des symboles d'éléments en utilisant la constante **Menu déroulant de symboles** dans la palette **Fonctions»Numérique» Constantes numériques supplémentaires**.

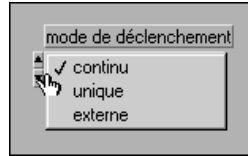
Si vous utilisez une valeur de -1 pour le symbole d'un élément, la liste déroulante affiche une ligne de division grisée au lieu d'afficher le nom de l'élément.

Commandes de type menu déroulant

Les menus déroulants sont des objets numériques spéciaux qui associent des valeurs numériques à des chaînes de caractères, à des images, ou aux deux. Les différents styles de menus déroulants sont représentés dans l'illustration suivante.



Les menus déroulants sont particulièrement utiles pour sélectionner des éléments mutuellement exclusifs, tels que des modes de déclenchement. Par exemple, vous pouvez souhaiter que les utilisateurs puissent choisir entre un déclenchement continu, un déclenchement unique et un déclenchement externe, comme cela est représenté dans l'illustration suivante.



Dans l'exemple précédent, le menu déroulant intitulé *Mode de déclenchement* a trois descripteurs de mode, listés l'un après l'autre, dans l'affichage du texte du menu déroulant. Le G dispose les éléments dans une liste circulaire (comme un Rolodex), un seul élément étant visible à la fois. Chaque élément a une valeur numérique, qui s'échelonne de zéro à $n - 1$, où n est le nombre d'éléments (trois dans cet exemple). Cependant, la valeur du menu déroulant peut être n'importe quelle valeur comprise dans sa gamme de données numériques. Il affiche le dernier élément (*Externe*, ci-dessus) pour n'importe quelle valeur supérieure ou égale à 2 et le premier élément (*Continu*, ci-dessus) pour n'importe quelle valeur inférieure ou égale à zéro.

Vous pouvez ouvrir le menu local et sélectionner **Visualiser»Afficheur numérique** pour afficher la valeur numérique associée à l'élément courant d'un menu déroulant.

Les utilisateurs peuvent utiliser ce menu déroulant pour choisir un élément à partir d'une liste facile à comprendre sans avoir à connaître la valeur qui représente cet élément. La valeur associée à l'élément sélectionné est transférée au diagramme, où, par exemple, vous pouvez sélectionner une condition à partir d'une structure Condition (code conditionnel) qui met à l'exécution l'élément sélectionné.

Vous pouvez sélectionner un élément dans une commande de menu déroulant de deux manières. Utilisez les boutons d'incréméntation pour passer à l'élément suivant ou au précédent. L'incréméntation continue de façon circulaire dans la liste des éléments tant que vous maintenez enfoncé le bouton de la souris. Vous pouvez aussi sélectionner n'importe quel élément directement en cliquant sur le menu déroulant avec l'outil Doigt, puis en choisissant l'élément que vous voulez dans le menu qui apparaît. Cependant, le menu déroulant simple fonctionne comme n'importe quel autre menu déroulant, sauf qu'il n'a pas de boutons d'incréméntation. Vous devez sélectionner un élément en cliquant sur le menu déroulant et en sélectionnant un élément dans le menu qui apparaît.

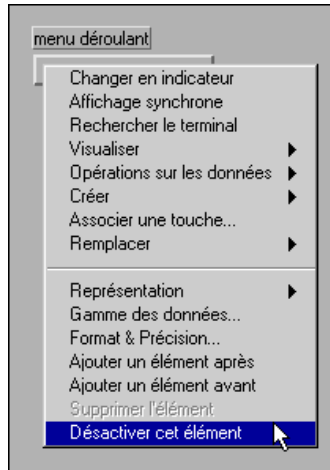
Ajouter des éléments de texte à des menus déroulants

Un nouveau menu déroulant de texte est composé d'un élément avec une valeur égale à zéro et un affichage contenant une chaîne de caractères vides. Vous pouvez entrer ou changer le texte dans la zone de texte du menu déroulant de la même manière que pour des étiquettes, à l'aide de l'outil Texte. Appuyez sur la touche <Entrée> (**Windows et HP-UX**), ou <Retour> (**Macintosh et Sun**) ou cliquez en dehors de la zone de texte pour finir la saisie. **Ajouter un élément après**, situé dans le menu local du menu déroulant de texte, crée un nouvel élément vide après l'élément actuel. **Ajouter un élément avant** insère un nouvel élément devant l'élément actuel. Si vous êtes par exemple en train d'éditer l'élément 0 lorsque vous sélectionnez **Ajouter un élément après**, un élément 1 est créé, prêt à recevoir le texte que vous entrez pour le nouvel élément. Vous pouvez également appuyer sur <Maj-Entrée> (**Windows et HP-UX**) ou <Maj-Retour> (**Macintosh et Sun**) après avoir tapé un élément pour passer à un nouvel élément.

Dans chaque menu déroulant, les valeurs des éléments situés au-dessus du point d'insertion augmentent d'une unité pour prendre en compte le nouvel élément. Par exemple, si vous insérez un élément après l'élément 4, le nouvel élément devient l'élément 5, l'élément qui était préalablement l'élément 5 devient le 6, le 6 devient le 7, et ainsi de suite. Si vous insérez un élément avant l'élément 4, le nouvel élément devient l'élément 4, l'élément qui était préalablement l'élément 4 devient le 5, le 5 devient le 6, et ainsi de suite.

Utilisez la commande **Supprimer l'élément** du menu local du menu déroulant pour supprimer un élément quelconque. Comme dans le cas d'un ajout d'élément, les valeurs numériques des éléments s'ajustent automatiquement.

Pour désactiver les éléments d'un menu déroulant pendant que vous êtes en cours d'édition, ouvrez son menu local et sélectionnez **Désactiver cet élément**, comme cela est représenté dans l'illustration suivante. Ensuite, sélectionnez l'élément que vous voulez désactiver.



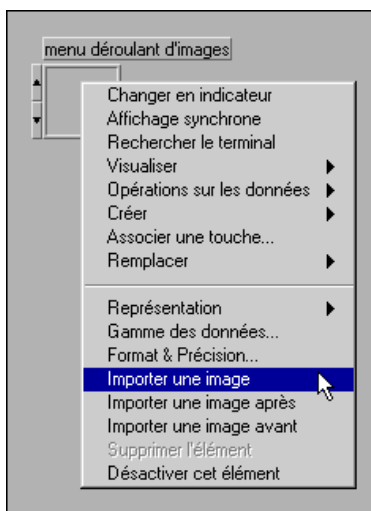
Si vous voulez activer un élément désactivé pendant que vous êtes en cours d'édition, ouvrez le menu local et sélectionnez **Activer cet élément**. Pour désactiver/activer un élément de menu déroulant à partir d'un programme,

utilisez l'élément de **Disabled Items** l'Attribute Node (qui est un tableau d'indices) représenté dans l'illustration suivante.



Ajouter des éléments d'images aux menus déroulants

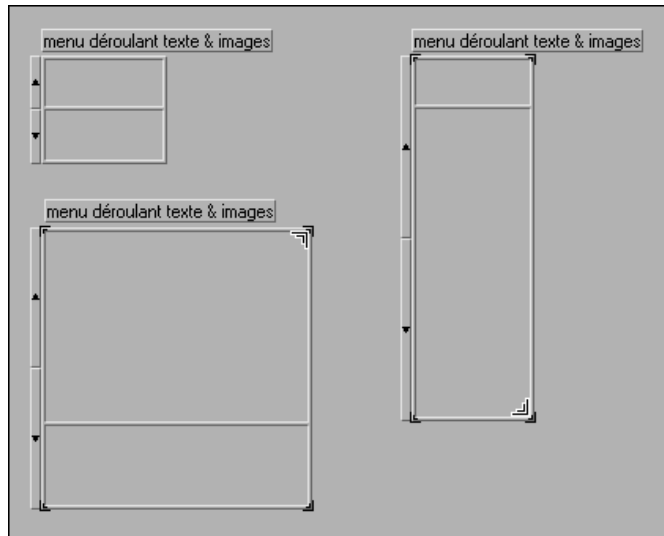
Un nouveau Menu déroulant d'images ou Menu déroulant texte & images est composé d'un élément avec une zone d'affichage d'image vide. Vous devez copier une image dans le Presse-papiers avant de pouvoir l'importer dans un Menu déroulant d'images. Lorsque vous avez une image dans le Presse-papiers, ouvrez le menu local du Menu déroulant d'images et sélectionnez **Importer une image**. Pour ajouter une autre image, copiez-la dans le Presse-papiers, puis ouvrez le menu local du Menu déroulant d'images et sélectionnez **Importer une image avant** ou **Importer une image après**, comme cela est représenté dans l'illustration suivante.



Changer la taille et le texte des menus déroulants texte & images

Vous pouvez agrandir le Menu déroulant texte & images pour laisser plus de place à son contenu, vous pouvez agrandir de la même manière toutes les commandes. Vous pouvez également réduire la taille. Lorsque vous redimensionnez le Menu déroulant texte & images à partir d'un de ses coins inférieurs, la hauteur de la zone de texte change. Lorsque vous redimensionnez le Menu déroulant texte & images à partir d'un de ses coins

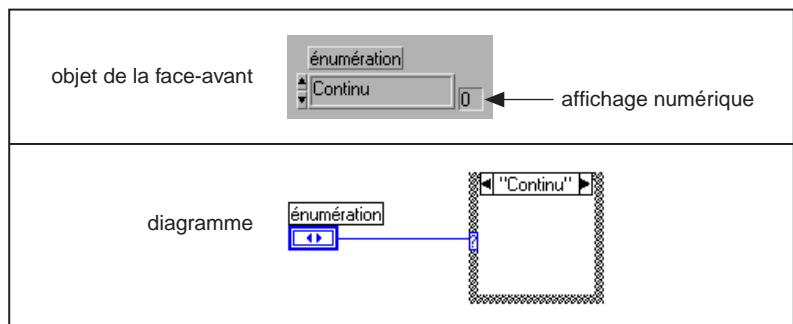
supérieurs, la hauteur de la zone d'image change. Les possibilités de redimensionnement sont représentées dans l'illustration suivante.



Commandes de type énuméré

Une commande de type énuméré est similaire à une commande de menu déroulant de texte. Cependant, si une donnée d'un type énuméré est connectée à une structure Condition, la condition affiche la chaîne de caractères, ou le texte de l'élément, au lieu d'un nombre. Le type des

données de type énuméré est octet non signé, mot non signé ou long mot non signé, sélectionnable à partir de la palette **Représentation**. Un exemple est représenté dans l'illustration suivante.



Un avantage d'utiliser des types énumérés au lieu de commandes de menu déroulant est que si vous utilisez un type énuméré sur le cadre connecteur d'un sous-VI, et que vous ouvrez ensuite un menu local pour créer une constante, une commande ou un indicateur, vous créez une énumération contenant les chaînes de caractères correctes.

Vous entrez des éléments dans un type énuméré de la même manière que vous entrez des éléments dans un menu déroulant, en utilisant **Ajouter un élément après** ou **Ajouter un élément avant**. Vous pouvez également utiliser l'outil Texte. Appuyez sur <Maj-Entrée> (**Windows et HP-UX**) ou <Maj-Retour> (**Macintosh et Sun**) pour entrer un nouvel élément. Cliquez sur n'importe quelle zone en dehors du type énuméré lorsque vous avez fini d'entrer des éléments.

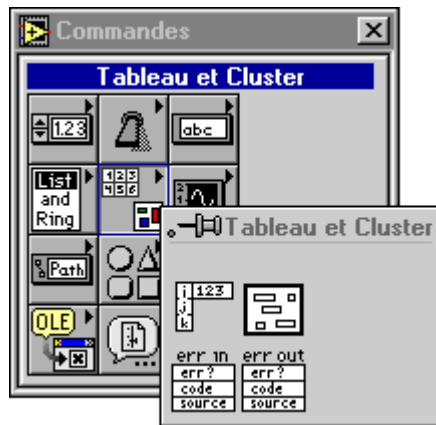
Si un type énuméré est câblé à une structure Condition, la structure Condition doit avoir des conditions qui gèrent chaque élément du type énuméré. Si la structure Condition a une condition par défaut, elle gère des éléments qui ne sont pas explicitement gérés par d'autres conditions.

Toutes les opérations arithmétiques, à l'exception de Incrément et Décrément, traitent le type énuméré de la même manière qu'une valeur numérique non signée. Incrément ajoute la dernière énumération à la première, et Décrément soustrait la première énumération à la dernière. Dans tous les autres cas, le type énuméré est traité comme une valeur non signée. Lorsqu'un entier signé est affecté à un type énuméré, les nombres négatifs sont affectés à la première énumération et les nombres positifs hors gamme à la dernière énumération. Les entiers non signés sont toujours affectés à la valeur de la dernière énumération.

Si vous connectez une valeur numérique à un indicateur de type énuméré, le nombre est converti en l'élément d'énumération le plus proche, les nombres hors gamme étant traités comme expliqué ci-dessus. Si vous connectez une commande d'énumération à une valeur numérique, la valeur est l'indice du type énuméré. Pour câbler un type d'énumération à un indicateur d'énumération, les éléments d'énumération doivent correspondre, bien que l'indicateur puisse avoir des éléments supplémentaires au-delà des éléments du type.

Commandes et indicateurs de tableau et de cluster

Ce chapitre présente la manière d'utiliser les tableaux et les clusters. Vous pouvez accéder aux tableaux et aux clusters par le biais de la palette **Commandes»Tableaux et Clusters**, présentée dans l'illustration suivante.



Consultez les exemples dans la bibliothèque `examples\general\arrays.llb`.

Vous pouvez utiliser **Aide»Référence en ligne»Function and VI Reference** pour découvrir des descriptions des fonctions G, la plupart s'exécutant sur des tableaux et des scalaires. Les rubriques *Fonctions des tableaux* et *Fonctions des clusters* décrivent les fonctions exclusivement conçues pour des opérations de tableau et de cluster.

Tableaux

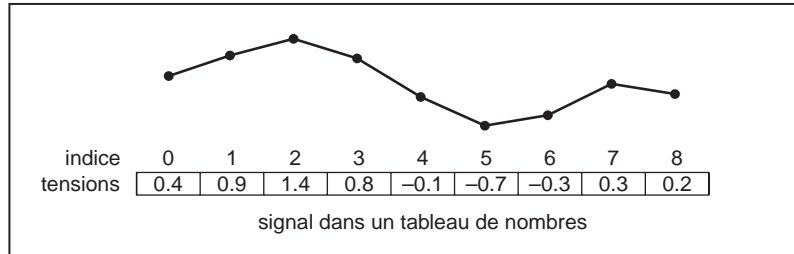
Un tableau est une série de taille variable d'éléments de données du même type, contrairement au cluster qui est une série de taille fixe d'éléments de données de types divers. Les éléments d'un tableau sont ordonnés et vous pouvez accéder à un élément particulier du tableau par une *indexation* du tableau. L'*indice* commence par zéro, ce qui signifie que sa gamme est comprise entre zéro et $n - 1$, où n est le nombre d'éléments du tableau, comme indiqué dans l'illustration suivante.

Indice	Tableau de 11 éléments
0	Yves
1	Louis
2	Pierre
3	François
4	Claire
5	Paul
6	Michel
7	Maurice
8	Etienne
9	Chantal
10	Albert

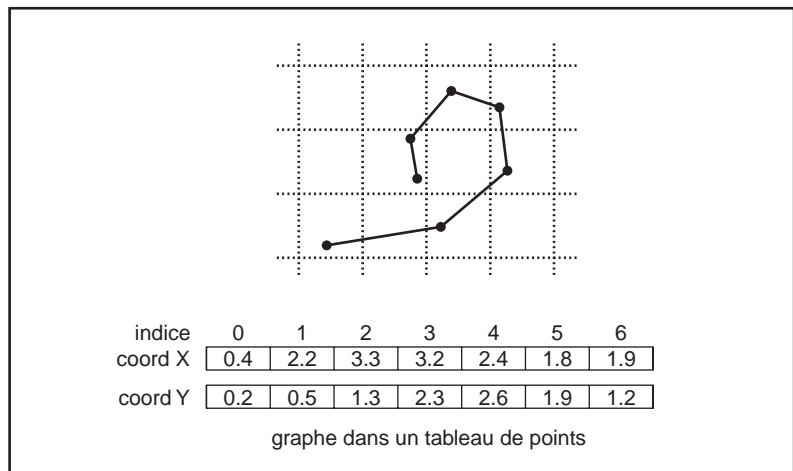
Un exemple simple de tableau peut être une liste de noms représentée en G comme un tableau de chaînes de caractères, comme indiqué dans l'illustration suivante.

Yves	Louis	Pierre	François	Claire	Paul	Michel	Maurice	Etienne	Chantal	Albert
Liste de noms dans un tableau de chaînes de caractères										

Un autre exemple peut être une forme d'onde représentée par un tableau numérique dans lequel chaque élément consécutif est une valeur de tension correspondant à un intervalle de temps, comme indiqué dans l'illustration suivante.

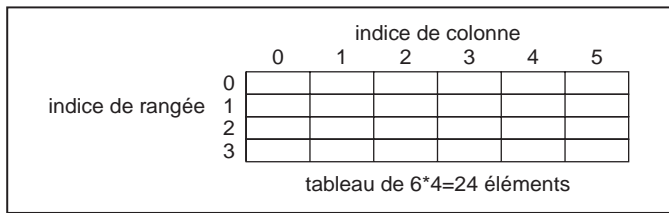


Un exemple plus complexe peut être un graphe constitué par un tableau de points où chaque point est une paire de nombres qui représentent les coordonnées X et Y, comme indiqué dans l'illustration suivante.



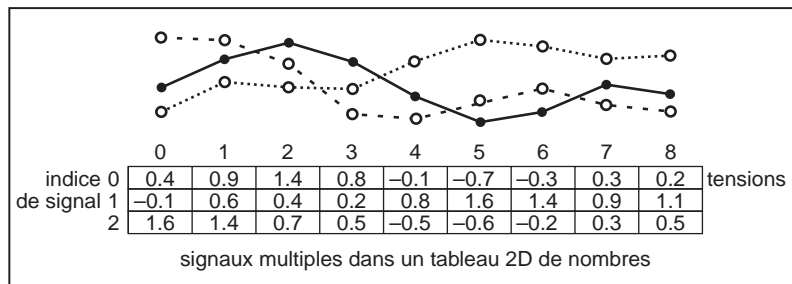
Les exemples précédents sont des tableaux à *une dimension* (1D). Un tableau à *deux dimensions* (2D) nécessite deux indices pour localiser un élément : un indice de colonne et un indice de ligne, les deux commençant

par zéro. Dans ce cas, on parle de tableau à N colonnes et M lignes qui contient N fois M éléments, comme indiqué dans l'illustration suivante.



Un exemple simple peut être un échiquier. Il comporte huit colonnes et huit lignes pour un total de 64 positions. Chacune d'entre elles peut être vide ou contenir une pièce. Vous pouvez donc représenter un échiquier comme un tableau de chaînes de caractères à deux dimensions. Chaque chaîne de caractères contient le nom de la pièce qui occupe l'emplacement correspondant sur l'échiquier, ou une chaîne de caractères vide si l'emplacement est vide. D'autres exemples familiers sont le calendrier, les horaires de train, et même les images de la télévision qui peuvent être représentées en tableaux de nombres à deux dimensions qui fournissent l'intensité de la lumière de chaque point. Les tableaux constituent un exemple familier aux utilisateurs d'ordinateur ; ils comprennent des lignes et des colonnes de nombres, des formules, et du texte.

Tous les exemples de tableaux à une dimension peuvent être généralisés en deux dimensions. Une série de courbes représentée dans un tableau de nombres à deux dimensions vous est proposée dans l'illustration suivante. L'indice de ligne désigne la courbe et l'indice de colonne indique le point sur la courbe.



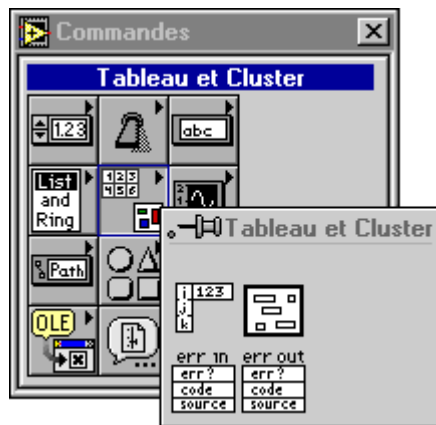
Les tableaux peuvent comporter un nombre de dimensions arbitraire, mais il faut un indice par dimension pour localiser un élément. Le type de données des éléments du tableau peut être un cluster contenant différents types, y compris des tableaux dont le type d'élément est un cluster, et ainsi

de suite. Vous ne pouvez pas avoir un tableau de tableaux. Par contre, vous pouvez utiliser un tableau multidimensionnel ou un tableau de clusters de tableaux.

Pensez à utiliser des tableaux lorsque vous travaillez avec une série de données similaires. Les tableaux sont souvent utiles lorsque vous exécutez des calculs répétitifs ou des fonctions d'E/S. L'utilisation de tableaux peut réduire la taille de votre application, l'accélérer et en faciliter le développement étant donné le grand nombre de fonctions de tableau et de VIs du G.

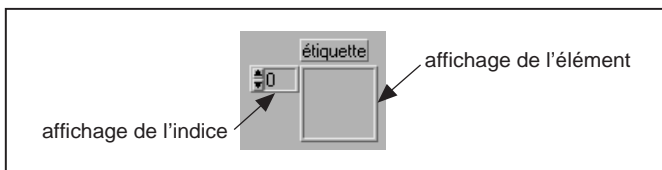
Création de commandes de tableau

Vous pouvez créer une commande de tableau en sélectionnant d'abord un tableau par le biais de **Commandes»Tableaux et Clusters**, comme indiqué dans l'illustration suivante.



Vous pouvez créer une commande ou un indicateur de tableau en combinant un *tableau vierge* de la palette **Tableaux et Clusters**, comme indiqué dans la précédente illustration et un *élément* valide, qui peut être un nombre, un booléen, une chaîne de caractères, un chemin, un refnum ou un cluster. L'élément ne peut pas être un autre tableau ou un graphe déroulant. Si l'élément est un graphe, alors seuls les types de données qui contiennent des clusters au lieu de tableaux au niveau principal sont valides.

La sélection d'un tableau dans la palette **Tableaux et Clusters** place un tableau vierge sur la face-avant, comme indiqué dans l'illustration suivante.



Un nouveau tableau vierge possède un *affichage d'indice*, un *affichage d'élément* vide et une étiquette optionnelle.

Il existe deux manières de définir un type de tableau. En faisant glisser une commande ou un indicateur du type voulu dans la fenêtre d'affichage de l'élément, ou en déposant la commande ou l'indicateur directement en ouvrant un menu local sur la face-avant, en sélectionnant la commande et en la faisant glisser dans l'affichage d'élément. Dans les deux cas, la commande ou l'indicateur que vous insérez comble l'affichage vide.

Par exemple, si vous voulez définir un tableau de booléens, ouvrez le menu local, sélectionnez une commande booléenne, et faites-la glisser dans l'affichage d'élément comme indiqué dans les figures 14-1 à 14-4.

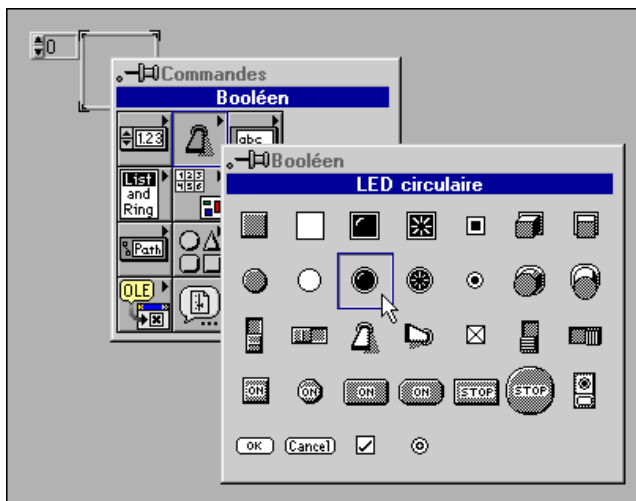


Figure 14-1. Ouvrez un menu local dans l'affichage d'élément et sélectionnez la commande Booléen.

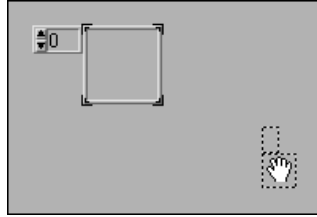


Figure 14-2. Relâchez le bouton de la souris. Le menu disparaît et le curseur se transforme en icône Défilement des fenêtres.

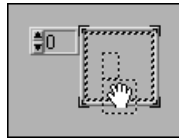
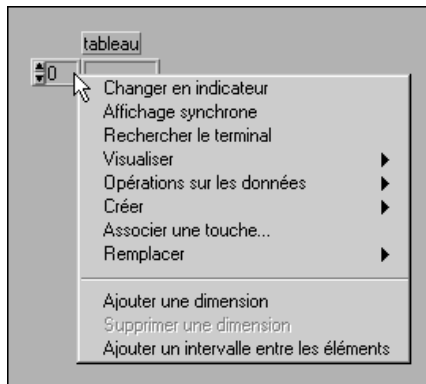


Figure 14-3. Faites glisser l'icône Défilement des fenêtres dans l'affichage d'élément.



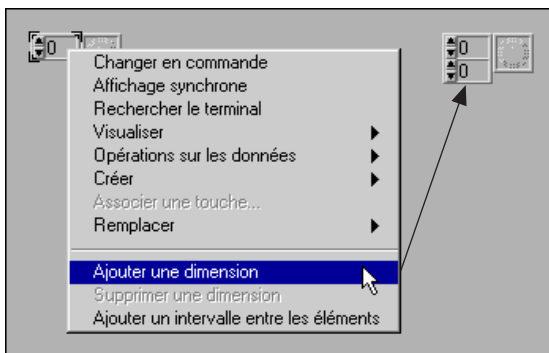
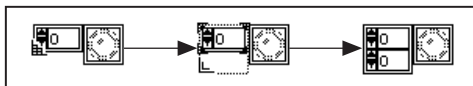
Figure 14-4. Cliquez sur le bouton de la souris. La commande est alors déposée dans l'affichage.

Le menu local de l'affichage de l'indice vous est présenté dans l'illustration suivante.



Dimensions du tableau

Un nouveau tableau possède une dimension et un affichage de l'indice. Vous pouvez redimensionner l'affichage de l'indice verticalement ou sélectionner l'élément **Ajouter une dimension** dans le menu local de l'affichage de l'indice pour ajouter une dimension. Vous pouvez rétrécir l'affichage d'indice verticalement ou sélectionner **Supprimer une dimension** pour la supprimer. Un affichage de l'indice supplémentaire apparaît pour chaque dimension ajoutée. Les deux méthodes de redimensionnement sont présentées dans les illustrations suivantes.

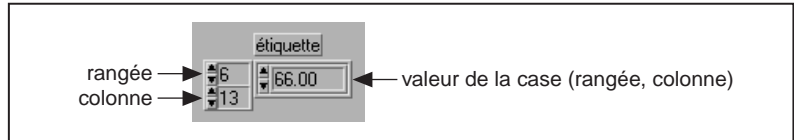


La dimension d'un tableau dépend du choix que vous effectuez lorsque vous décidez de la quantité d'identificateurs nécessaires à la localisation d'un élément de données. Par exemple, pour localiser un mot dans un livre, vous iriez au sixième mot de la vingt-huitième ligne de la page 192. Trois indices, 6, 28, et 192 spécifient le mot. Une des représentations possibles du livre serait un tableau de mots à trois dimensions (3D). Pour retrouver un livre dans une bibliothèque, vous indiquez la position sur l'étagère, l'étagère elle-même, la bibliothèque, l'allée et l'étage. Cette représentation dans un tableau comprendrait donc cinq dimensions. La localisation d'un mot dans une bibliothèque nécessiterait alors huit indices.

Remarquez que lorsque vous ajoutez une dimension à un tableau, le fil change de style, d'épaisseur et de couleur. Pour plus d'informations, consultez le chapitre 18, *Câblage du diagramme*.

Affichage de l'indice de tableau

Si vous interprétez un tableau à deux dimensions en termes de lignes et de colonnes, l'affichage du haut serait l'indice de ligne et celui du bas serait l'indice de colonne. L'affichage combiné à droite indique la valeur de la position spécifiée, comme indiqué dans l'illustration suivante.



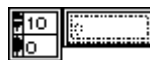
Par exemple, supposons que vous ayez un tableau de 3 lignes et de 4 colonnes avec les valeurs de l'illustration suivante.

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

Les lignes et les colonnes commencent par zéro, ce qui signifie que la première colonne est la colonne 0, la deuxième est la colonne 1, et ainsi de suite. La modification de l'affichage de l'indice en ligne 1, colonne 2 affiche une valeur de 6, comme indiqué dans l'illustration suivante.



Si vous essayez d'afficher quelque chose qui se trouve en dehors des limites, l'affichage de la valeur est alors grisé pour indiquer qu'aucune valeur n'est actuellement définie. Par exemple, si vous essayez d'afficher un élément de la ligne 10 du précédent tableau, l'affichage est grisé, comme indiqué dans l'illustration suivante.





Outil Coin de tableau



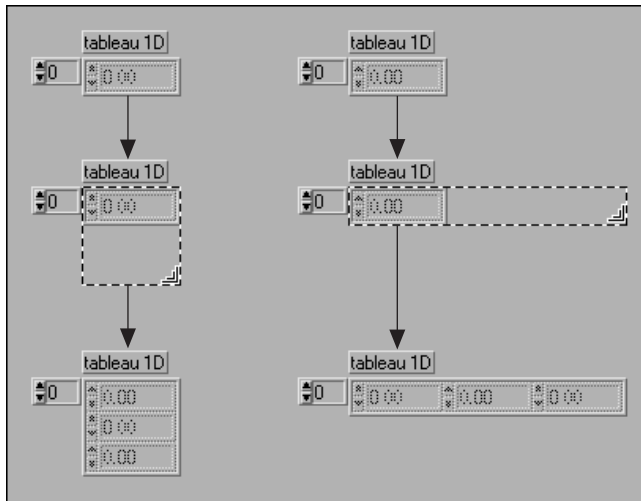
Symbole normal de l'outil Coin

Affichage de tableaux sous forme élément unique ou tabulaire

Un nouveau tableau s'affiche sous forme d'élément unique. Le tableau affiche la valeur d'un seul élément : celui auquel l'affichage de l'indice fait référence. Vous pouvez également afficher le tableau sous forme de table d'éléments en redimensionnant le squelette du tableau par le biais d'un des quatre coins qui encadrent l'élément. L'outil Coin de tableau diffère légèrement de l'outil Coin normal lorsqu'il est placé au-dessus d'une zone de redimensionnement du tableau vierge et le curseur redevient normal lorsque vous commencez à redimensionner. Un exemple vous est présenté dans l'illustration suivante.



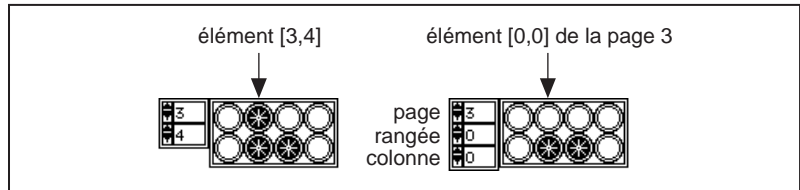
L'illustration suivante montre la manière de redimensionner verticalement ou horizontalement un tableau à une dimension afin d'afficher une plus grande quantité d'éléments en même temps.



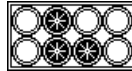
L'affichage de l'indice contient l'indice de l'élément situé à gauche ou dans le coin supérieur gauche de la table. En modifiant l'indice, vous pouvez afficher différentes sections d'un tableau de grande taille ou multidimensionnel. Pour des tableaux à une dimension, l'indice identifie la colonne de l'élément visible à l'extrême gauche. Pour des tableaux à deux dimensions ou plus, les deux indices du bas identifient les coordonnées de

l'élément visible en haut à gauche. Dans la partie gauche de l'illustration suivante, cet élément est [3,4].

Si vous considérez un tableau à trois dimensions comme un livre de pages composées de lignes (lignes) de caractères (colonnes), le tableau affiche une partie ou l'intégralité d'une page. La figure de droite de l'illustration suivante affiche les quatre premières colonnes des deux premières lignes de la page trois du tableau.



Pour masquer les indices du tableau, ouvrez un menu local sur le cadre extérieur et désactivez l'élément **Visualiser»Affichage de l'indice**. L'illustration suivante montre un exemple de tableau tabulaire sans affichage de l'indice.



Pour afficher un tableau dans une autre orientation que celle du format de tableau tabulaire, affichez les éléments dans un cluster sur la face-avant et traitez-les dans un tableau. Pour cela, utilisez les fonctions “Tableau en cluster” et “Cluster en tableau” présentées dans la rubrique **Référence en ligne»Function and VI Reference»Cluster Function**.

Manipulation des tableaux

Vous pouvez manipuler un élément d'un tableau de la même manière que s'il n'en faisait pas partie. Vous pouvez manipuler un affichage de l'indice de la même manière qu'une commande numérique. Pour définir la valeur d'un élément de tableau de commande, affichez l'élément avec l'affichage de l'indice et définissez ensuite la valeur.

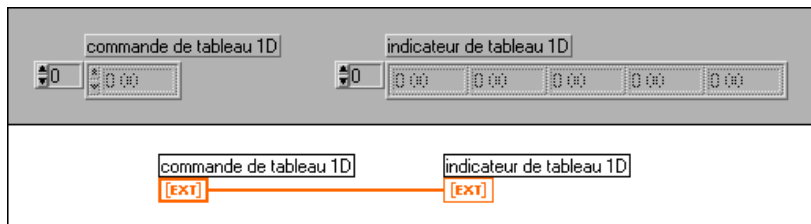
Taille et valeurs par défaut des tableaux

Vous ne pouvez pas limiter la taille d'un tableau par un nombre d'éléments fixe. Toutefois, lorsque vous définissez les valeurs par défaut d'une commande de tableau, vous pouvez également définir la taille par défaut. *N'attribuez pas une taille par défaut plus importante que celle dont vous*

avez besoin. En effet, toutes les données par défaut sont enregistrées avec le VI et augmentent par conséquent la taille de votre fichier.

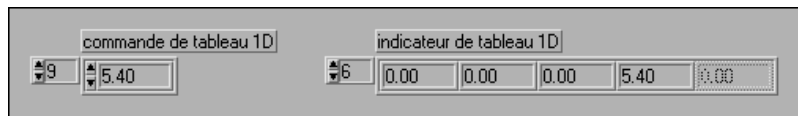
Un nouveau tableau vierge sans élément est *non-défini*. Il ne possède pas de type de données ni d'éléments et ne peut pas être utilisé dans un programme. Une fois que vous avez déposé une commande dans le tableau, c'est un *tableau vide* (sa longueur est de 0) avec le type défini de la commande déposée. Bien qu'il n'ait pas d'éléments définis, vous pouvez tout de même l'utiliser dans un VI. La commande de tableau affiche tous les éléments en gris clair, indiquant qu'ils ne sont pas définis.

Les illustrations suivantes présentent une commande de tableau sous forme d'élément unique liée à un indicateur de tableau sous forme tabulaire avec cinq éléments affichés. La première figure montre que les deux tableaux sont vides (les valeurs des éléments ne sont pas définies).



Si vous définissez l'affichage de l'indice de la commande de tableau à 9 et la valeur de l'élément d'indice 9 à 5,4, les données du tableau augmentent à 10 éléments (de 0 à 9). La valeur de l'élément 9 est la valeur que vous définissez (5,4 dans cet exemple). Le G attribue aux autres éléments la valeur par défaut de la commande numérique, 0,00. Si vous sélectionnez à présent **Prendre les valeurs actuelles par défaut** soit dans le squelette soit dans le menu local de la commande de tableau, les données par défaut du

tableau comprennent les 10 éléments spécifiés. Une fois le VI exécuté, l'indicateur affiche les mêmes valeurs.



Le fait de sélectionner **Réinitialiser avec les paramètres par défaut** pour un élément réinitialise cet élément uniquement avec sa valeur par défaut.

Maintenant, supposons que vous modifiez la valeur par défaut d'un élément numérique de 0,0 à 1,0. Effectuez cette opération en modifiant la valeur de l'indicateur avec l'outil Doigt, en ouvrant ensuite un menu local sur l'élément et en sélectionnant **Prendre les valeurs actuelles par défaut**. Supposons également que vous définissiez la valeur de l'élément 12 à 7,0. Le tableau possède maintenant 13 éléments. Les valeurs des dix premiers éléments ne changent pas, mais les éléments 10 et 11 ont une nouvelle valeur par défaut de 1,0, et l'élément 12 se voit attribuer la valeur de 7,0. Cet exemple vous est présenté dans l'illustration suivante.



Si vous sélectionnez à présent **Réinitialiser avec les paramètres par défaut** soit avec le menu local du squelette ou de l'indice, le tableau revient à 10 éléments, comme celui décrit dans l'exemple précédent.

Pour augmenter la taille d'un tableau de $[N_i \text{ par } N_j \text{ par } N_k \dots]$ à $[M_i \text{ par } M_j \text{ par } M_k \dots]$, attribuez une valeur au dernier nouvel élément $[M_i - 1, M_j - 1, M_k - 1 \dots]$. Pour réduire la taille, exécutez d'abord la commande **Opérations sur les données»Vider le tableau** par le biais du menu local d'indice du tableau. Ensuite, définissez le tableau et les valeurs à la taille que vous voulez, ou sélectionnez les sous-ensembles non voulus du tableau

et sélectionnez l'élément **Opérations sur les données»Couper des données** dans le menu local. Cette opération vous est expliquée dans les sections suivantes.

Éléments du tableau

En mode Exécution, utilisez la touche <Tab> pour déplacer le focus de la touche soit entre les commandes de la face-avant soit entre les éléments à l'intérieur d'un tableau. Au départ, la touche <Tab> déplace le focus de la touche entre les commandes de la face-avant. Pour le déplacer entre les éléments d'un tableau spécifique, effectuez d'abord un <Tab> vers ce tableau. Ensuite, utilisez la touche <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) et la flèche orientée vers le bas pour déplacer le focus de la touche à l'intérieur du tableau. Vous pouvez à présent utiliser la touche <Tab> pour passer en revue les éléments et les

indices du tableau. Pour revenir au passage entre les commandes par la touche <Tab>, utilisez la touche <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) et la flèche orientée vers le haut.

Recherche de la taille des tableaux

Pour trouver la taille d'une commande ou d'un indicateur de tableau, sélectionnez **Opérations sur les données**»**Visualiser le dernier élément** dans le menu local d'indice du tableau.

Déplacement ou redimensionnement des tableaux

Déplacez toujours le tableau en cliquant sur le bord du squelette ou sur l'affichage de l'indice puis en le faisant glisser. Si vous faites glisser l'élément dans le tableau, il se sépare du tableau, le squelette et les éléments n'étant pas verrouillés les uns aux autres. Si vous faites malencontreusement glisser un élément alors que vous vouliez déplacer le squelette, annulez l'opération en refaisant glisser l'élément dans le squelette, ou au-delà du bord de la fenêtre de la face-avant, ou en utilisant la touche <Echap> avant de relâcher le bouton de la souris. Vous pouvez annuler une opération de redimensionnement en faisant glisser le bord au-delà de la fenêtre de la face-avant avant de relâcher le bouton de la souris. Si le tableau fait partie de la sélection en cours, le fait de saisir un élément fait glisser le tableau.

Redimensionnez verticalement l'indice du tableau à partir de n'importe quel coin, ou horizontalement à partir de la gauche.

Sélection des cellules du tableau

Pour copier des données d'un tableau ou y coller des données, vous devez d'abord sélectionner une zone du tableau. Vous pouvez copier ou coller des données en effectuant les opérations suivantes.

1. Pour sélectionner les données, mettez l'indice du tableau sur le premier élément de l'ensemble de données que vous voulez copier.
2. Sélectionnez ensuite l'élément **Opérations sur les données» Commencer la sélection** dans le menu local du tableau.
3. Puis, mettez l'indice du tableau sur le dernier élément de l'ensemble de données que vous voulez copier.
4. Sélectionnez l'élément de menu **Opérations sur les données» Terminer la sélection** dans le menu local du tableau.
5. Enfin, sélectionnez **Opérations sur les données» Copier** ou **Opérations sur les données» Coller** pour copier ou coller les éléments sélectionnés.

En mode Exécution, ces éléments de menu sont directement disponibles lorsque vous ouvrez un menu local sur un tableau. Le processus de sélection est détaillé dans l'exemple suivant.

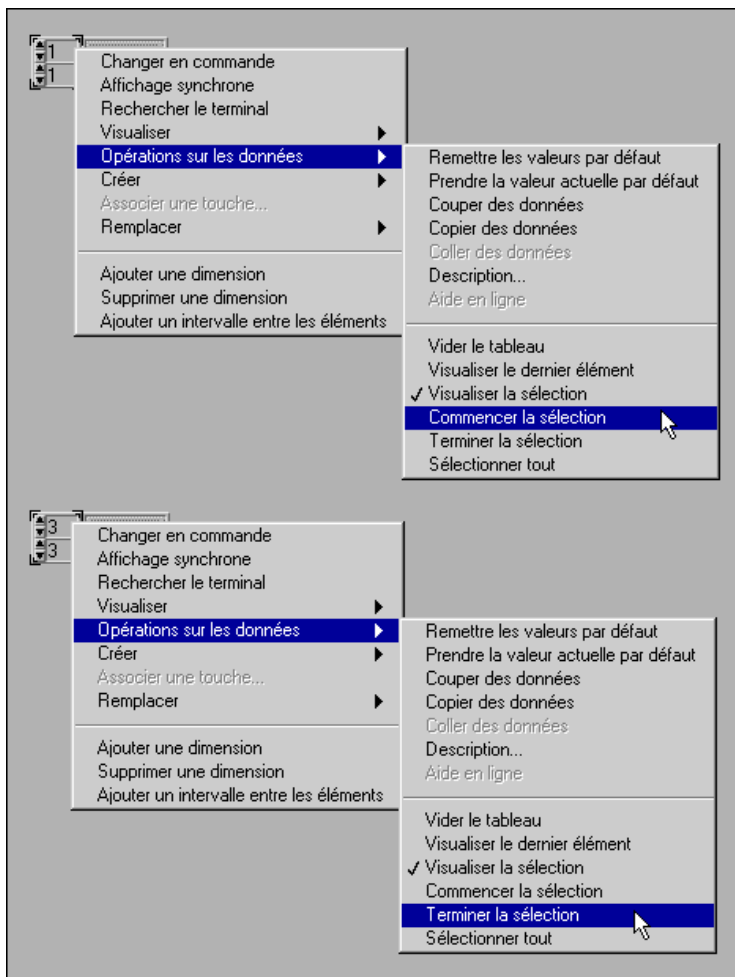
Un exemple de sélection de cellules d'un tableau

Commencez par sélectionner **Visualiser la sélection** dans le menu local pour activer l'affichage des cellules sélectionnées.

Un cadre entoure les cellules sélectionnées. Choisissez **Opérations sur les données» Ajouter un intervalle entre les éléments** dans le menu local. Cette opération sépare les cellules par un petit espace ouvert. Bien qu'il ne soit pas nécessaire d'ajouter un intervalle entre les éléments, ce cadre ouvert change légèrement et améliore l'apparence du tableau. Lorsque vous sélectionnez des cellules, cet espace ouvert est comblé par un cadre épais, ce qui marque les cellules sélectionnées. Si vous n'ajoutez pas d'intervalle entre les éléments, le cadre de sélection s'affiche par-dessus la bordure des cellules sélectionnées.

Vous pouvez définir les cellules de la sélection par le biais de l'indice du tableau. L'exemple suivant utilise un tableau à deux dimensions.

Définissez l'indice du tableau à 1, 1. Choisissez **Commencer la sélection** dans le menu local. Définissez l'indice du tableau à 3, 3. Choisissez **Terminer la sélection** dans le menu local.



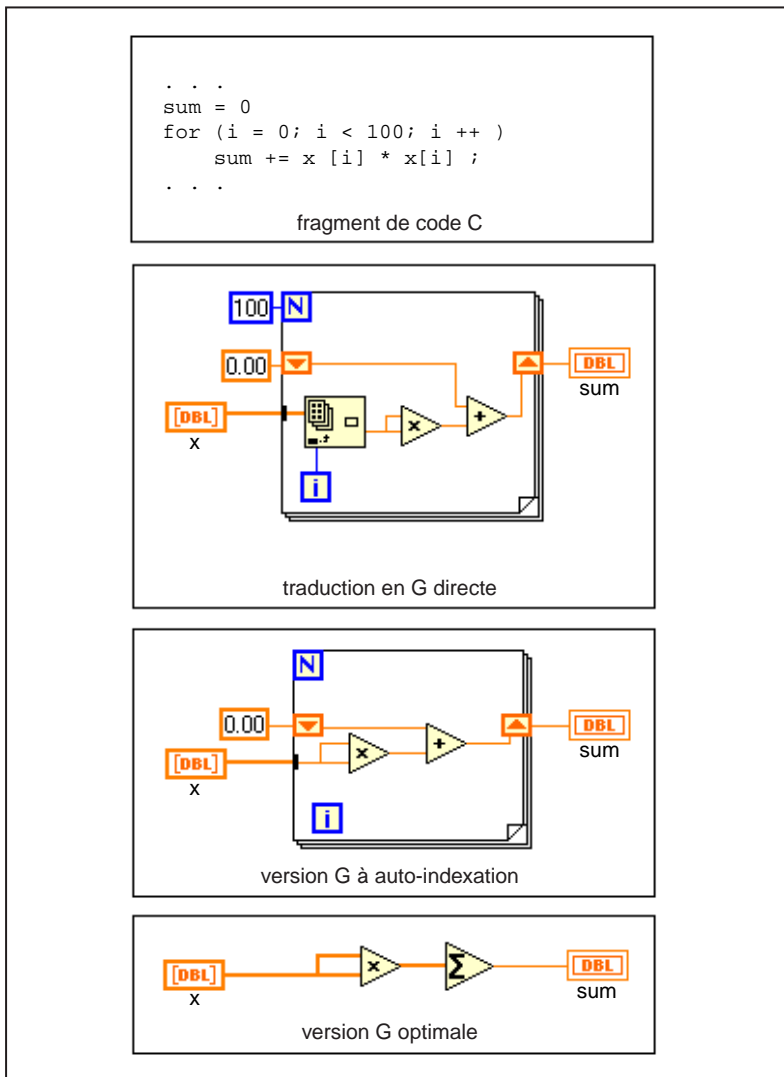
La sélection comprend les cellules désignées par des indices de valeur plus faible et non pas celles désignées par des indices de valeur plus élevée. Si vous voulez inclure les cellules de 1, 1 à 3, 3, définissez l'indice à 4, 4 avant de choisir **Terminer la sélection**. Remarquez que le fait de couper les éléments sélectionnés supprime l'ensemble des lignes et des colonnes comprises dans la sélection, mais ne place que la zone sélectionnée dans le Presse-papiers.

Une fois que vous avez sélectionné des cellules, coupez ou copiez les données à coller dans d'autres cellules. Remarquez qu'une ligne du cadre reste en surbrillance lorsque vous avez terminé. C'est un point d'insertion et il reste dans le tableau. Pour masquer cette ligne, désélectionnez l'élément **Visualiser la sélection** dans le menu local. Vous pouvez éliminer cette ligne sans désélectionner **Visualiser la sélection** en vidant la sélection. Définissez l'indice à 0 dans toutes les dimensions, et sélectionnez **Commencer la sélection** et **Terminer la sélection**. Cette opération définit la sélection de 0 à 0 dans toutes les dimensions, ce qui correspond à la sélection par défaut lorsque vous placez le tableau sur la face-avant.

Tableaux G et tableaux d'autres systèmes

La plupart des langages de programmation possèdent des tableaux, mais rares sont ceux qui intègrent les fonctions de tableau G. Généralement, les autres langages se contentent des opérations de tableau de base qui consistent à extraire ou à remplacer un élément et laissent aux utilisateurs le soin de concevoir des opérations plus complexes. Le G possède ces opérateurs de base et vous pouvez donc intégrer un programme écrit dans un autre langage directement dans un VI G. Toutefois, vous pouvez généralement créer un diagramme plus simple et de plus petite taille si vous redémarrez le G et utilisez les fonctions de tableau de niveau principal.

L'exemple suivant montre un fragment d'un programme en C qui additionne les carrés des éléments d'un tableau, la conversion directe dans un diagramme G et deux versions plus efficaces que la conversion.

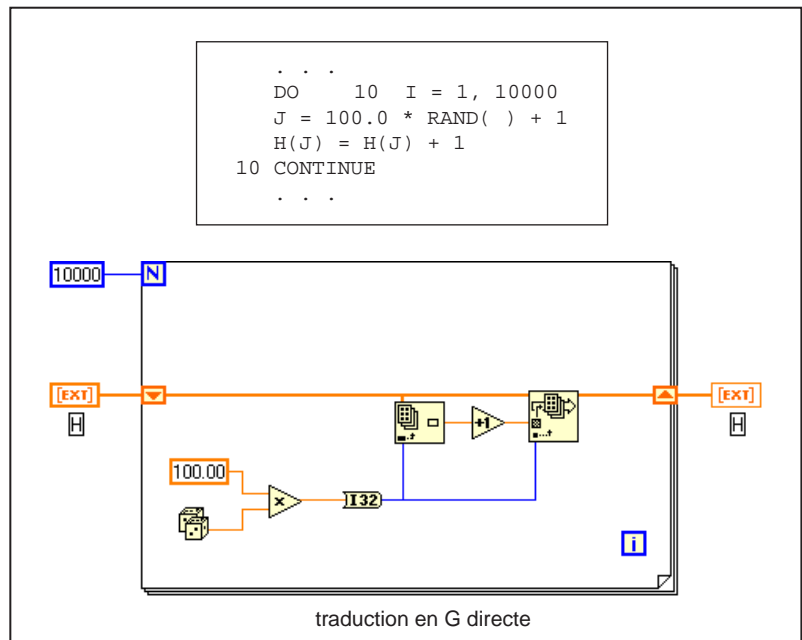




Icône de fonction
"Indexer un
tableau"

L'opération d'indexation pour l'extraction d'un élément est représentée en C par des crochets [] qui suivent le nom du tableau et encadrent l'indice. En G, cette opération est représentée par l'icône de fonction "Indexer un tableau" illustrée à gauche. Le tableau est lié au terminal supérieur gauche, l'indice est lié au terminal inférieur gauche, et la valeur de l'élément du tableau est liée au terminal de droite.

L'exemple suivant montre l'opération de base qui consiste à remplacer un élément du tableau. Le fragment de code FORTRAN génère un histogramme de nombres aléatoires (le générateur de nombre aléatoire produit des valeurs comprises entre 0 et 1), comme le fait la version G. Ne tenez pas compte du +1 dans le calcul FORTRAN de J ; l'addition est nécessaire car les tableaux FORTRAN commencent l'indexation avec le chiffre plutôt que zéro.



Fonction "Remplacer
un élément d'un
tableau"

L'opération d'indexation pour l'insertion ou le remplacement d'un élément est représentée en FORTRAN par des parenthèses () à gauche du signe = (égal) qui suit le nom du tableau, et qui encadrent l'indice. En G, cette opération est représentée par l'icône de la fonction "Remplacer un élément d'un tableau", illustrée à gauche. Le tableau est lié au terminal supérieur gauche, la valeur de remplacement est liée au terminal gauche du milieu, l'indice est lié au terminal inférieur gauche et le tableau mis à jour est lié au terminal de droite.

Avant de pouvoir utiliser un tableau dans la plupart des langages de programmation, vous devez d'abord le déclarer et, dans certains cas, l'initialiser. En G, la conception d'un indicateur ou d'une commande de tableau sur la face-avant revient à déclarer le tableau. La définition d'une valeur par défaut pour ce dernier revient à lui donner une valeur de départ.

Dans la déclaration, il faut généralement indiquer à la plupart des langages la taille maximale d'un tableau. Les informations concernant la taille du tableau ne sont pas traitées de manière intégrée au tableau, ce qui signifie que vous devez en assurer le suivi vous-même. En G, il n'est pas nécessaire de déclarer une taille maximale des tableaux, le G conservant automatiquement les informations sur la taille du tableau. De plus, le G refuse de stocker une valeur qui dépasse les limites du tableau. Cette vérification n'existe pas dans la plupart des langages de programmation. Vous pourriez donc détériorer la mémoire par mégarde.

Clusters

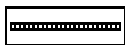
Un cluster G constitue une série ordonnée d'un ou de plusieurs éléments, identique aux structures C et à d'autres langages. Contrairement aux tableaux, les clusters peuvent contenir toutes les combinaisons de types de données G. Bien que les éléments de cluster et de tableau soient tous deux ordonnés, vous avez accès aux éléments du cluster en *désassemblant* tous les éléments en une seule fois plutôt qu'en indexant un élément à la fois. Les clusters sont également différents des tableaux dans la mesure où ils sont de taille fixe. Comme un tableau, un cluster peut être soit une commande soit un indicateur ; un cluster ne peut pas contenir un mélange de commandes et d'indicateurs.

Vous pouvez utiliser des clusters pour grouper des éléments de données apparentés qui se situent à différents emplacements du diagramme. Cette opération réorganise les fils et réduit le nombre de terminaux connecteurs dont les sous-VIs ont besoin.

La plupart des clusters du diagramme ont un motif de fil commun, même si les clusters de nombres (parfois appelés *points*) possèdent un motif de fil spécial.



Motif de fil de cluster commun



Motif de fil de cluster de nombres

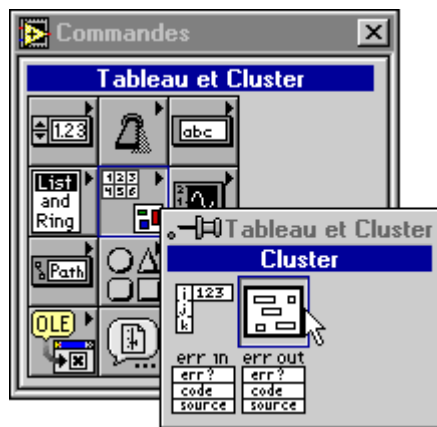
Vous ne pouvez connecter des terminaux que s'ils possèdent le même type. Pour les clusters, cela signifie que les deux clusters doivent avoir le même nombre d'éléments et que les éléments correspondants – déterminés par l'ordonnement du cluster – doivent être du même type. Le G contraint les nombres de différentes représentations dans le même type.

L'ordonnement du cluster est un ordonnancement numérique associé aux éléments d'un cluster. Il est décrit dans la section *Ordonnement des éléments du cluster* plus loin dans ce chapitre.

Pour plus d'informations sur les fonctions qui gèrent les clusters, consultez la rubrique **Référence en ligne»Function and VI ReferenceVI»Cluster Function**.

Création des clusters

Vous pouvez créer une commande ou un indicateur de type cluster en installant n'importe quelle combinaison de booléens, de chaînes de caractères, de graphes déroulants, de graphes scalaires, de tableaux ou même d'autres clusters dans un *squelette de cluster*. Vous pouvez accéder au squelette du cluster à partir de la palette **Commandes**, comme indiqué dans l'illustration suivante.



Un nouveau squelette de cluster possède une bordure redimensionnable et une étiquette optionnelle.

Lorsque vous ouvrez un menu local dans une zone d'élément vide, la palette **Commandes** s'affiche. Vous pouvez placer n'importe quel élément de la palette **Commandes** dans un cluster. Vous pouvez faire glisser des objets existants dans le squelette du cluster ou les créer et les déposer

directement à l'intérieur en les sélectionnant dans le menu local de la commande. Le cluster prend la direction de données (commande ou indicateur) du premier élément que vous placez dans le cluster, ainsi que les objets ajoutés par la suite. En sélectionnant **Changer en commande** ou **Changer en indicateur** dans le menu local de n'importe quel élément du cluster, vous changez le cluster et tous ses éléments d'indicateurs en commandes ou inversement.

Manipulation et configuration des éléments du cluster

A l'exception de la définition des valeurs par défaut et des éléments de menu **Changer en commande** et **Changer en indicateur**, vous pouvez configurer les commandes et les indicateurs de la même manière que ceux qui ne se trouvent pas dans un cluster.

Éléments de cluster

En mode Exécution, utilisez la touche <Tab> pour déplacer le focus de la touche soit entre les commandes de la face-avant soit entre les éléments à l'intérieur d'un cluster. Au départ, la touche <Tab> déplace le focus de la touche entre les commandes de la face-avant. Pour le déplacer entre les éléments d'un cluster spécifique, effectuez d'abord un <Tab> vers ce cluster. Ensuite, utilisez la touche <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) et la flèche orientée vers le bas pour vous déplacer entre les éléments du cluster. Pour revenir au passage entre les commandes par la touche <Tab>, utilisez la touche <Ctrl> (**Windows**), <commande> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) et la flèche orientée vers le haut.

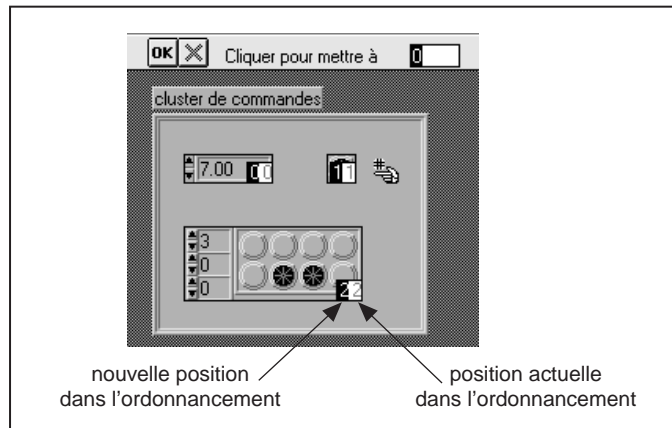
Valeurs par défaut du cluster

Pour définir la valeur par défaut de l'ensemble du cluster pour la valeur en cours de chaque élément, ouvrez un menu local sur le cadre du cluster et sélectionnez la commande **Opérations sur les données»Prendre les valeurs actuelles par défaut**. Pour réinitialiser tous les éléments en leur configuration individuelle par défaut, sélectionnez la commande **Opérations sur les données»Réinitialiser les valeurs par défaut** dans le menu local du cluster.

Pour modifier la valeur par défaut d'un seul élément dans un cluster, ouvrez un menu local sur cet élément et sélectionnez **Prendre les valeurs actuelles par défaut**.

Ordonnement des éléments du cluster

Les éléments du cluster ont un ordonnancement logique qui n'a pas de rapport avec leur position dans le squelette. Le premier objet que vous insérez dans le cluster est l'élément 0, le deuxième est l'élément 1, et ainsi de suite. Si vous supprimez un élément, l'ordonnement est automatiquement ajusté. Vous pouvez modifier l'ordonnement en cours en sélectionnant l'élément **Ordonner le cluster...** dans le menu local du cluster. L'apparence de l'élément change alors, comme indiqué dans l'illustration suivante.



Les zones blanches sur les éléments indiquent leur emplacement dans l'ordonnement du cluster. Les zones noires indiquent leur nouvelle position. En cliquant sur un élément avec le curseur d'ordonnement du cluster, vous pouvez définir la position de l'élément dans le cluster par le biais du nombre affiché dans la barre d'outils. Modifiez cet ordonnancement en tapant une nouvelle valeur dans ce champ et cliquez sur la commande correspondante. Une fois que l'ordonnement vous satisfait, cliquez sur le bouton Entrée pour le définir et quittez le mode d'édition de l'ordonnement du cluster. Cliquez sur le bouton X pour retourner à l'ancien ordonnancement.



Bouton Entrée



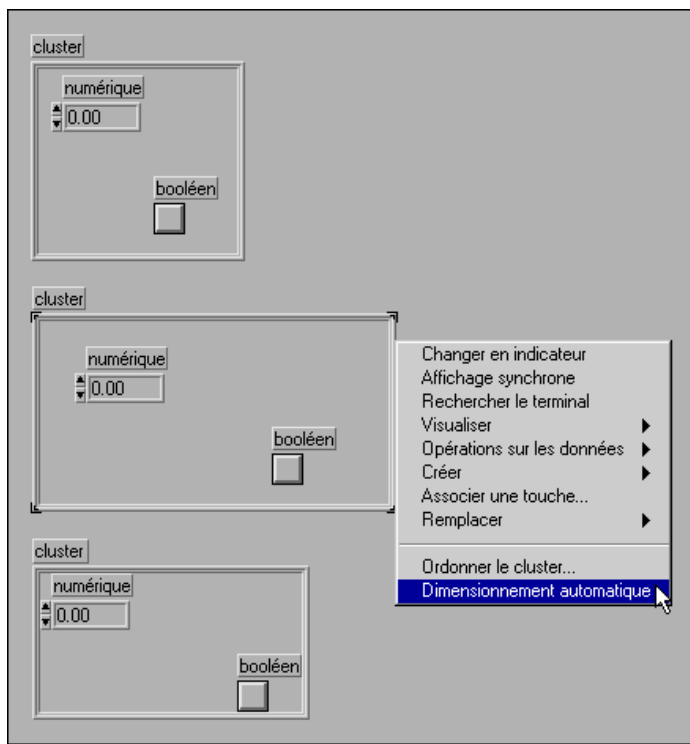
Bouton X

L'ordonnement du cluster détermine l'ordre des éléments apparaissant sous forme de terminaux dans les fonctions Assembler et Désassembler du diagramme. Pour plus d'informations sur ces fonctions, consultez le sujet **Référence en ligne»Fonction and VI Reference»Cluster Functions**.

Déplacement ou redimensionnement des clusters

Les éléments du cluster ne sont pas des composants fixes. En effet, vous pouvez les déplacer ou les redimensionner indépendamment même s'ils se situent dans le squelette. Pour éviter de les sortir du squelette par mégarde, cliquez sur le squelette lorsque vous voulez déplacer un cluster et redimensionnez les clusters à partir du bord du squelette. Si malgré tout vous déplacez un élément alors que vous vouliez déplacer l'intégralité du cluster, vous pouvez toujours annuler l'opération en refaisant glisser l'élément dans le squelette ou au-delà du bord de la face-avant ou en appuyant sur la touche <Echap> avant de relâcher le bouton de la souris. Vous pouvez également annuler une opération de redimensionnement en faisant glisser le bord au-delà de la fenêtre de la face-avant ou en appuyant sur la touche <Echap> avant de relâcher le bouton de la souris.

Vous pouvez rétrécir les clusters afin que leur taille s'ajuste à leur contenu en sélectionnant **Dimensionnement automatique** dans le menu local, comme présenté dans l'illustration suivante.



Assemblage des clusters

Le G possède trois fonctions pour l'assemblage ou la construction des clusters. Les fonctions Assembler et "Assembler par nom" assemblent un cluster à partir d'éléments séparés ou remplacent des éléments séparés par des éléments du même type. La fonction "Tableau en cluster" convertit un tableau de plusieurs éléments en cluster d'éléments.

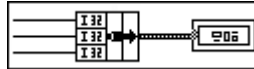
Fonction Assembler



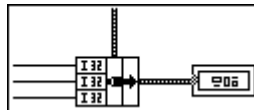
Fonction
Assembler

La fonction Assembler, à laquelle vous pouvez accéder par le biais de la palette **Tableaux et Clusters** de la palette **Fonctions**, s'affiche sur le diagramme avec deux terminaux d'entrée de type élément dans la partie gauche. Redimensionnez l'icône verticalement pour créer autant de terminaux que nécessaire. L'élément que vous liez au terminal du haut devient l'élément 0 dans le cluster de sortie. L'élément que vous liez au deuxième terminal devient l'élément 1, et ainsi de suite.

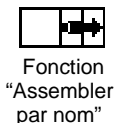
Au fur et à mesure que vous connectez à chaque terminal d'entrée un symbole représentant le type de données de l'élément associé s'affiche dans le terminal. Ce dernier vous est présenté dans l'illustration suivante. Vous devez lier toutes les entrées que vous créez.



En plus des terminaux d'entrée des éléments dans la partie gauche, la fonction Assembler possède également un terminal d'entrée de clusters central, comme indiqué dans l'illustration suivante. Vous pouvez utiliser ce terminal pour remplacer un ou plusieurs éléments d'un cluster existant sans affecter les autres éléments.



Consultez la section *Remplacement d'éléments de clusters* de ce chapitre pour un exemple d'utilisation de cette fonction.



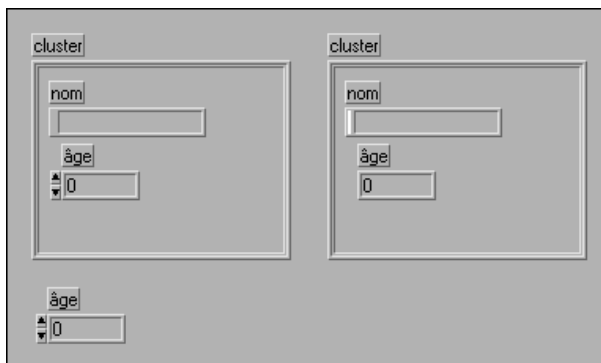
Fonction "Assembler par nom"

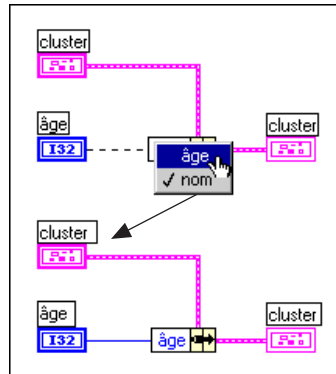
La fonction "Assembler par nom", qui se trouve également dans la palette **Fonctions»Cluster**, est identique à la fonction Assembler. Toutefois, au lieu de référencer les champs par leur position, vous les référencez par leur nom. Contrairement à la fonction Assembler, vous ne pouvez afficher que les éléments auxquels vous voulez accéder. Pour chaque élément auquel vous voulez accéder, vous devez ajouter une entrée à la fonction.

Etant donné que le nom ne dénote pas une position dans un cluster, vous devez également lier un cluster au terminal central. Vous pouvez utiliser la fonction "Assembler par nom" pour ne remplacer que les éléments par nom, mais pas pour créer un nouveau cluster. Toutefois, vous pourriez lier un type de données cluster au terminal central, et ensuite lier toutes les nouvelles valeurs sous forme de noms afin d'obtenir le même résultat.

Par exemple, supposons que vous ayez un cluster contenant une chaîne de caractères appelée Nom et un nombre appelé Age. Après avoir placé la fonction "Assembler par nom", vous devez d'abord connecter un cluster d'entrée au terminal central de la fonction "Assembler par nom".

Une fois le terminal central connecté, sélectionnez les éléments que vous voulez modifier en ouvrant un menu local sur un des terminaux de gauche de la fonction "Assembler par nom". Vous voyez apparaître une liste des noms des éléments du cluster source. Après avoir sélectionné un nom, vous pouvez lier une nouvelle valeur à ce terminal pour attribuer une nouvelle valeur à cet élément du cluster. Un exemple de cette opération vous est présenté dans les illustrations suivantes.



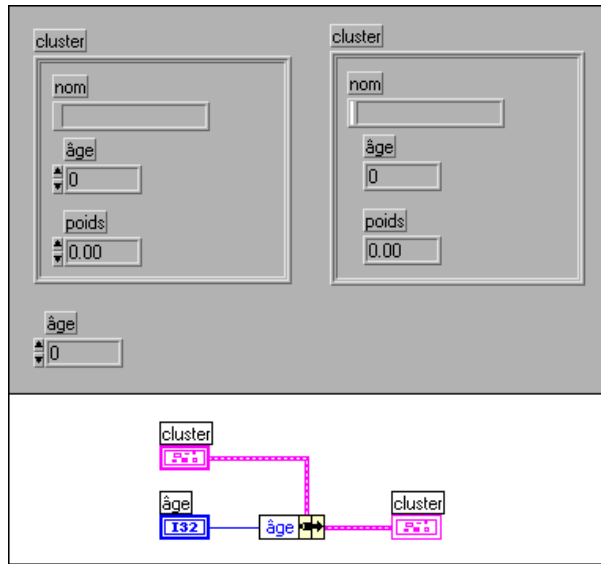


Redimensionnez la fonction “Assembler par nom” pour afficher autant d’éléments que vous voulez. Cette caractéristique est différente de la fonction Assembler. En effet, cette dernière nécessite que vous la dimensionniez afin qu’elle ait le même nombre d’éléments que le cluster produit.

Si vous ouvrez un menu local sur un seul élément de la fonction “Assembler par nom” lorsque le cluster contient un autre cluster, vous y trouverez un sous-menu latéral, **Sélectionner les éléments**, que vous pouvez utiliser pour accéder à chaque élément du sous-cluster à partir du nom, ou pour sélectionner tous les éléments.

La fonction “Assembler par nom” se révèle utile lorsque vous travaillez avec des structures de données qui peuvent changer pendant le processus de développement. Si votre modification implique le rajout d’un nouvel élément, ou un réordonnement du cluster, elle ne nécessite alors pas de modification au niveau de la fonction “Assembler par nom” car les noms sont toujours valides. Par exemple, si vous avez modifié l’exemple précédent en ajoutant un nouvel élément et “poids” aux clusters de source

et de destination, le VI reste toujours correct. Cet exemple vous est présenté dans l'illustration suivante.



La fonction “Assembler par nom” est particulièrement utile dans de plus grandes applications conjointement avec des définitions de type. Si vous définissez un cluster qui peut changer de définition de type, vous pourrez alors modifier tout VI qui utilise cette commande afin de prendre en compte le nouveau type de données. Pour cela, vous devez mettre à jour le fichier de la définition de type. Si les VIs n'utilisent que les fonctions “Assembler par nom” et “Désassembler par nom” pour accéder aux éléments du cluster, vous pouvez éviter la séparation des VIs qui utilisent ce cluster.

Pour plus d'informations sur les définitions de type, consultez la section *Définitions de type* du chapitre 24, *Commandes personnalisées et définitions de type*. De même, consultez la section *Fonction “Séparer par nom”* de ce chapitre.

Fonction “Tableau en cluster”



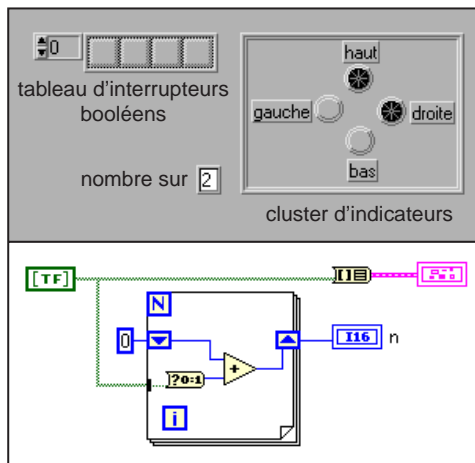
La fonction “Tableau en cluster”, à laquelle vous accédez par le biais des palettes **Cluster** ou **Tableau** de la palette **Fonctions**, convertit les éléments d'un tableau 1D en éléments d'un cluster. Cette fonction est utile lorsque vous voulez afficher des éléments du même type à l'intérieur d'un indicateur de cluster de face-avant, mais que vous voulez également manipuler les éléments par valeur d'indice sur le diagramme.

Utilisez l'élément **Taille du cluster...** du menu local de la fonction pour définir le nombre d'éléments dans le cluster. L'illustration suivante présente la boîte de dialogue qui s'affiche.



La fonction attribue l'élément du tableau *zéro* à l'élément du cluster *zéro*, et ainsi de suite. Si le tableau contient plus d'éléments que le cluster, la fonction ignore les éléments restants. Si le tableau contient moins d'éléments que le cluster, la fonction attribue aux éléments du cluster supplémentaires la valeur par défaut pour le type de données de l'élément.

L'illustration suivante présente la manière d'afficher un tableau de données dans un indicateur de cluster de face-avant. Grâce à cette technique, vous pouvez organiser les éléments du tableau de la face-avant afin de les adapter à votre application. Vous pourriez utiliser un indicateur de tableau tabulaire à la place, mais il vous limite à un format d'affichage fixe – horizontal avec le plus petit élément à gauche et un affichage d'indice attaché. La boucle illustrée compte simplement le nombre d'éléments ayant la valeur TRUE dans le tableau.



Séparation des clusters

Le G possède trois fonctions pour la séparation des clusters. Les fonctions Séparer et “Séparer par nom” séparent un cluster en éléments séparés. La fonction “Cluster en tableau” convertit un cluster d’éléments de type identique en un tableau d’éléments de ce type.

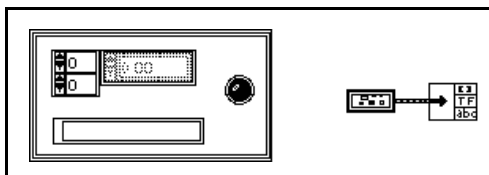
Fonction Séparer



Fonction
Séparer

La fonction Séparer, à laquelle vous pouvez accéder par le biais de la palette **Tableaux et Clusters** de la palette **Fonctions**, possède deux terminaux de sortie d’élément dans la partie droite. Vous pouvez ajuster le nombre des terminaux avec l’outil Coin de la même manière que vous ajustez la fonction Assembler. L’élément 0 de l’ordonnancement du cluster est transmis au terminal de sortie du haut, l’élément 1 est transmis au deuxième terminal, et ainsi de suite.

Le fil du cluster reste brisé jusqu’à la création du nombre exact de terminaux de sortie. Une fois cette opération effectuée, le fil devient continu. Les symboles du terminal affichent les types de données de l’élément, comme indiqué dans l’illustration suivante.



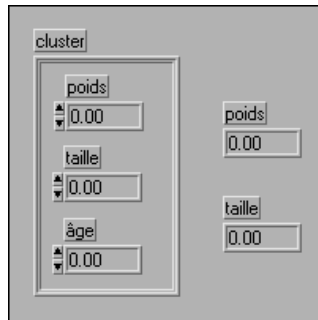
Fonction “Séparer par nom”



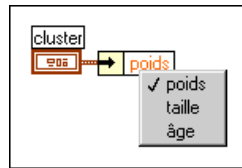
Fonction
“Séparer
par nom”

La fonction “Séparer par nom”, qui se situe également dans la palette **Tableaux et Clusters** de la palette **Fonctions**, est identique à la fonction Désassembler. Toutefois, au lieu de référencer les champs par leur position, la fonction les référence par leur nom. Contrairement à la fonction Séparer, vous n’affichez que les éléments que vous voulez lire. Il n’est pas nécessaire d’avoir une sortie pour chaque élément du cluster.

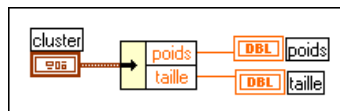
Par exemple, l’illustration suivante présente un panneau avec un cluster de trois éléments : le poids, la taille et l’âge.



Lorsque vous êtes connecté à une fonction “Séparer par nom”, vous pouvez ouvrir un menu local sur un terminal de sortie de la fonction. Ensuite, vous pouvez choisir **Sélectionner un élément** pour sélectionner un élément que vous voulez lire parmi les noms des composants du cluster, comme indiqué dans l’illustration suivante.

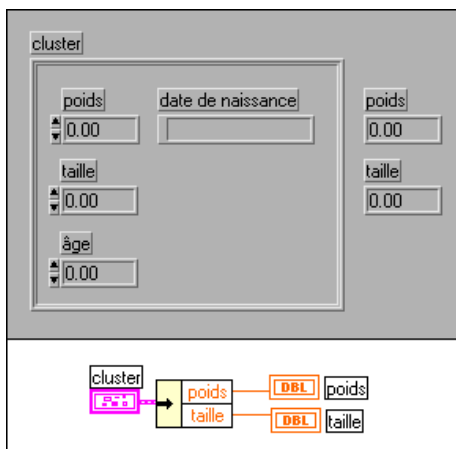


Redimensionnez la fonction pour lire d’autres éléments, comme indiqué dans l’illustration suivante. Souvenez-vous que vous n’avez pas besoin de lire tous les éléments et que vous pouvez les lire dans n’importe quel ordre.



Un des avantages de la fonction “Séparer par nom” par rapport à la fonction Séparer réside dans le fait qu’elle n’est pas aussi liée à la structure des données du cluster. La fonction Assembler doit toujours avoir le même nombre de terminaux que d’éléments. Si vous ajoutez ou supprimez un élément d’un cluster connecté à une fonction Séparer, vous brisez des fils. Grâce à la fonction “Séparer par nom”, vous pouvez ajouter ou supprimer des éléments du cluster sans rompre le VI, tant que les éléments ne sont pas référencés sur le diagramme.

Par exemple, avec le cluster précédent, vous pouvez ajouter une chaîne de caractères pour la date de naissance et le diagramme reste correct. Cet exemple vous est présenté dans l’illustration suivante.



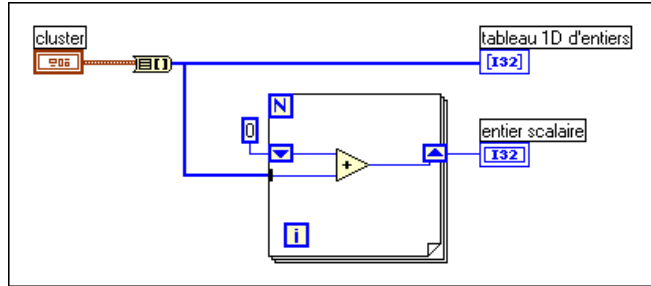
Comme la fonction “Assembler par nom”, “Séparer par nom” se révèle particulièrement utile dans de grandes applications conjointement avec des définitions de type. Pour plus d’informations, consultez la section *Fonction “Assembler par nom”* de ce chapitre et la section *Définitions de type* du chapitre 24, *Commandes personnalisées et définitions de type*.

Fonction “Cluster en tableau”



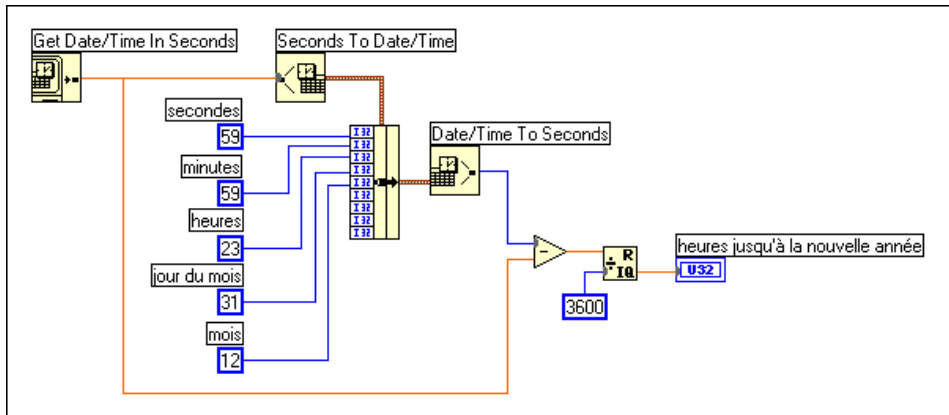
La fonction “Cluster en tableau”, à laquelle vous accédez par le biais de la palette **Conversion** de la palette **Fonctions**, convertit les éléments d’un cluster en un tableau 1D de ces éléments. Les éléments du cluster doivent tous être du même type. Le tableau contient le même nombre d’éléments que le cluster.

La fonction “Cluster en tableau” est utile lorsque vous voulez grouper un ensemble de commandes de face-avant dans un ordre particulier à l’intérieur d’un cluster et que vous voulez également les traiter comme un tableau sur le diagramme, comme indiqué dans l’illustration suivante.



Remplacement d'éléments de clusters

Vous devez parfois remplacer ou modifier la valeur d'un ou de deux éléments d'un cluster sans affecter les autres. Pour cela, désassemblez un cluster et liez directement les éléments inchangés à une fonction Assembler accompagnés des valeurs de remplacement de l'autre élément. L'exemple suivant propose une méthode plus pratique. Cet exemple calcule le nombre d'heures restant jusqu'au nouvel an en utilisant le cluster date-heure. Etant donné que le terminal d'entrée du cluster (terminal central) de la fonction Assembler est connecté, les seuls terminaux d'entrée d'élément qui doivent être câblés sont ceux qui possèdent des valeurs de remplacement.

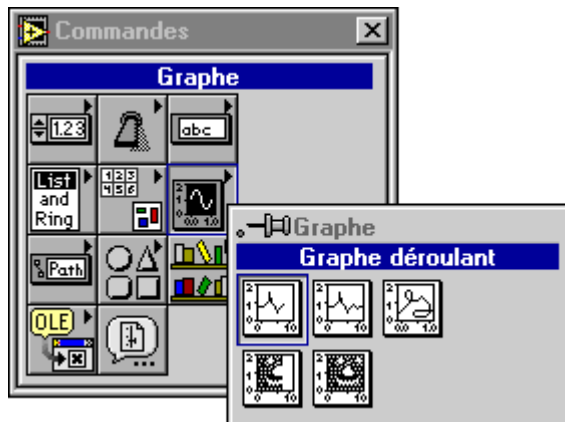


Indicateurs et commandes de graphe et de graphe déroulant

Ce chapitre présente la manière de créer et d'utiliser les indicateurs de graphe et de graphe déroulant de la palette **Commandes»Graphe**.

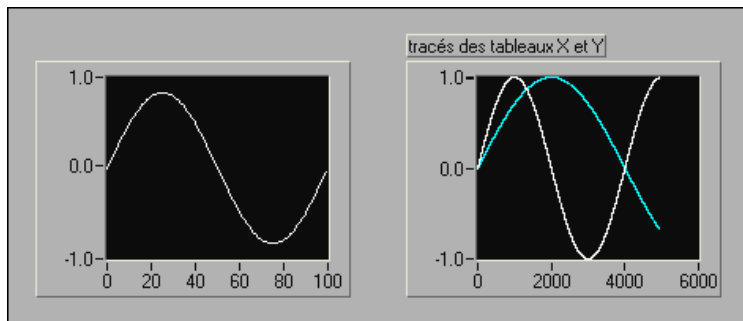
Un indicateur de graphe est un affichage en deux dimensions (2D) d'un ou de plusieurs *tracés*. Le graphe reçoit et trace les données en un bloc. Un graphe déroulant affiche également des tracés, mais il reçoit les données et met à jour l'affichage point par point ou tableau par tableau, et conserve un certain nombre de points précédents dans un buffer dédié à l'affichage. En complément des informations contenues dans ce chapitre, nous vous conseillons d'étudier des exemples de graphe et de graphe déroulant. Vous en trouverez dans le répertoire `examples\general\graphs`.

Le G possède trois types de graphes et deux types de graphes déroulants. Ils vous sont présentés dans l'illustration suivante en commençant par le haut et de gauche à droite : graphe déroulant, graphe oscilloscopique, graphe XY, graphe déroulant d'intensité et graphe d'intensité.



Création de forme d'onde et de graphes XY

Les *graphes oscilloscopiques* sont destinés à l'affichage de mesures échantillonnées de manière régulière. Les *graphes XY* peuvent afficher tout un ensemble de points, qu'ils soient régulièrement échantillonnés ou non. Vous pouvez obtenir un graphe oscilloscopique et un graphe XY par le biais de la palette **Commandes»Graphe**. Des exemples de ces graphes vous sont proposés dans l'illustration suivante.



Le graphe oscilloscopique ne trace que des fonctions à valeur unique avec des points répartis de manière régulière le long de l'axe des X, telles que des acquisitions de formes d'ondes continues qui varient en fonction du temps. Le graphe XY est polyvalent. C'est un objet graphique cartésien que vous pouvez utiliser pour tracer des fonctions à valeurs multiples telles que des formes circulaires ou des formes d'ondes à base de temps variable. Les deux graphes peuvent afficher un nombre de tracés illimité.

Chacun de ces graphes accepte différents types de données. Vous pouvez ainsi afficher vos données sans devoir les manipuler. Ces types de données sont décrits dans la section suivante.

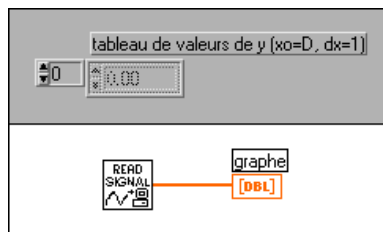
Vous trouverez d'autres sections qui décrivent quelques-unes des options avancées de graphe à la suite de la section sur les types de données de graphe : personnalisation de l'apparence des graphes, affichage et manipulation des curseurs et affichage d'une légende des graphes.

Graphes à tracé unique

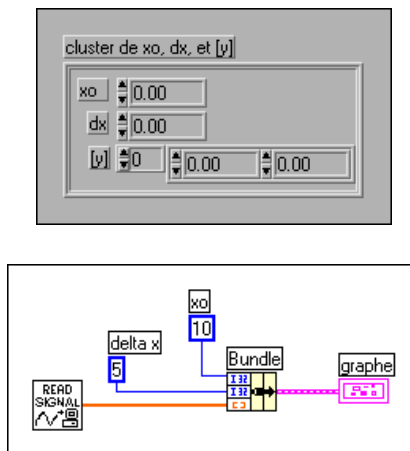
Types de données des graphes oscilloscopiques

Pour des graphes à tracé unique, le graphe oscilloscopique accepte deux types de données. Ces types de données sont décrits dans les paragraphes suivants.

Le premier type de données accepté par le graphe oscilloscopique est un simple tableau de valeurs. Le graphe interprète les données comme des points et les incrémente d'une unité, en commençant par $x = 0$. Le diagramme suivant illustre la manière dont vous créez ce type de données.



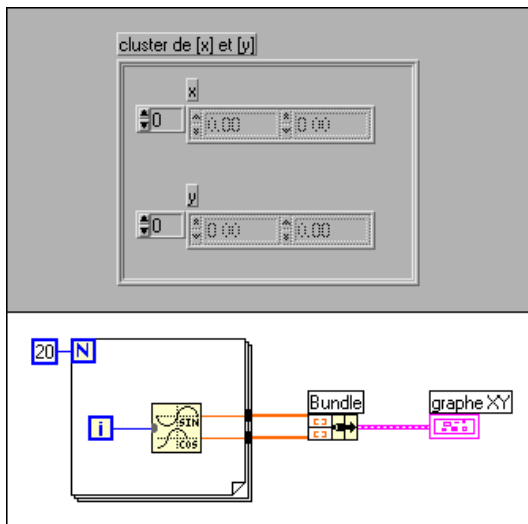
Le deuxième type de données est un cluster avec une valeur initiale x , une valeur Δx , et un tableau de données y . Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



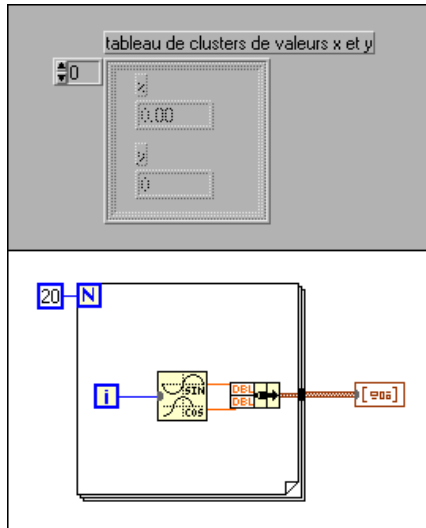
Types de données des graphes XY

Le graphe XY accepte deux types de données pour un graphe à tracé unique. Ces types de données sont décrits dans les paragraphes suivants.

Le premier type de données accepté par le graphe XY est un cluster contenant un tableau x et un tableau de données y . Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



Le deuxième type de données est un tableau de *points*, où un point est un cluster d'une valeur x et d'une valeur y . Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



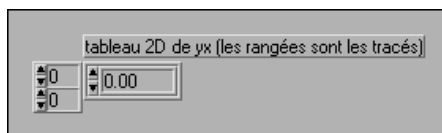
Graphe déroulant multicourbes

Vous pouvez afficher plusieurs tracés sur un graphe oscilloscopique ou un graphe xy unique. Dans l'ensemble, vous devez utiliser des tableaux des types de données décrits dans la précédente section pour définir l'affichage de plusieurs tracés dans un graphe unique. Etant donné que le G n'accepte pas les tableaux de tableaux, vous pouvez utiliser soit des tableaux 2D soit des tableaux de clusters de tableaux dans le cas où une création de tableau d'un type de données de tracé unique produit un tableau de tableaux.

Types de données des graphes oscilloscopiques

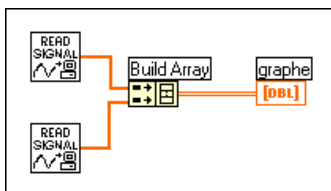
Le graphe oscilloscopique multicourbes accepte cinq types de données, décrits dans les paragraphes suivants.

Le premier type de données accepté par le graphe oscilloscopique multicourbes est un tableau de valeurs à deux dimensions où chaque ligne de données est un tracé unique. Le graphe interprète ces données comme des points, en commençant par $x = 0$ et en incrémentant x de un.

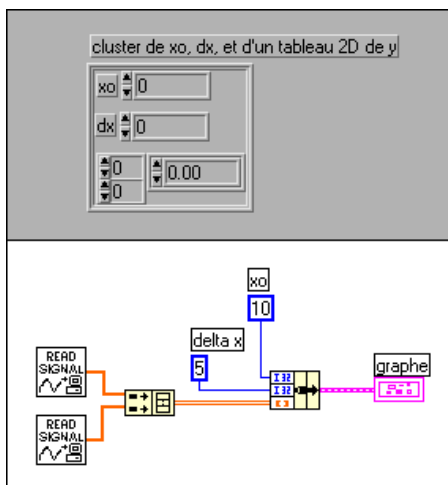


Lorsque l'élément **Transposer le tableau** du menu local du graphe est sélectionné, il interprète chaque colonne de données comme un tracé. Cette fonction est particulièrement utile lors de l'échantillonnage de plusieurs voies par une carte d'acquisition de données. En effet, ces données sont transformées en tableaux 2D, avec chaque voie stockée dans une colonne séparée.

L'exemple suivant vous indique comment utiliser un graphe pour afficher deux signaux où chaque signal est une ligne séparée d'un tableau à deux dimensions.



Le deuxième type de données est un cluster avec une valeur initiale x , une valeur Δx , et un tableau de données y . Les données y sont interprétées de manière identique que pour le précédent type de données. Ce type de données est utile pour l'affichage de plusieurs signaux tous échantillonnés à la même fréquence. Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.

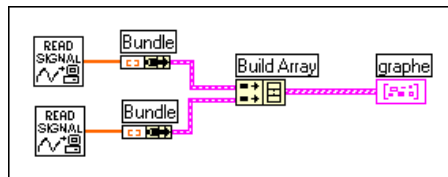


Le troisième type de données est un tableau de clusters d'un tableau de données y . Le tableau interne décrit les points d'un tracé et le tableau externe possède un élément pour chaque tracé.



Utilisez cette structure de données au lieu d'un tableau à deux dimensions si le nombre d'éléments dans chaque tracé est différent. Par exemple, utilisez-la pour échantillonner des données de plusieurs voies sur des durées différentes. Vous pouvez utiliser cette structure de données car chaque ligne d'un tableau à deux dimensions doit comporter le même nombre d'éléments, alors que le nombre d'éléments des tableaux internes d'un tableau de clusters peut varier.

L'illustration suivante vous indique comment créer la structure de données appropriée à partir de deux tableaux.



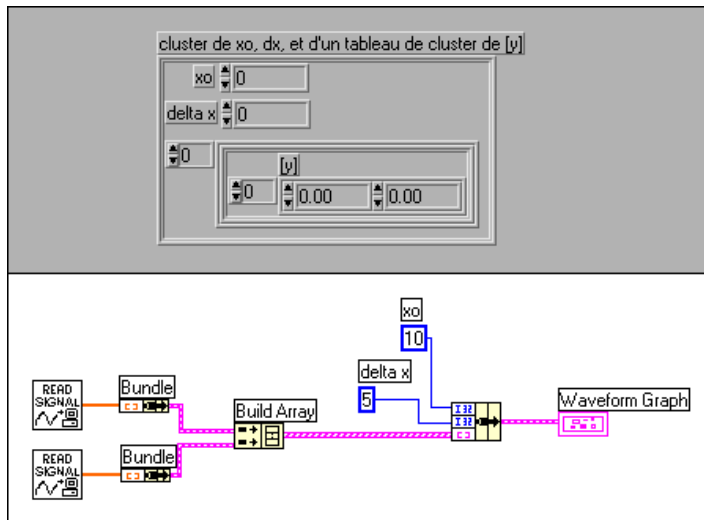
Pour obtenir un exemple de ce concept, observez le waveform Graph VI de la bibliothèque `examples\general\graphs\gengraph.11b`.

Une autre manière de convertir des tableaux en éléments d'un tableau de clusters consiste à utiliser la fonction "Construire un tableau de clusters".

Le quatrième type de données est un cluster avec une valeur initiale x , une valeur Δx , et un tableau de clusters d'un tableau de données y .

Utilisez cette structure de données au lieu d'un tableau à deux dimensions si le nombre d'éléments dans chaque tracé est différent. Par exemple, utilisez-la pour échantillonner des données de plusieurs voies sur des durées différentes. Vous pouvez utiliser cette structure de données car chaque ligne d'un tableau à deux dimensions doit comporter le même nombre d'éléments, alors que le nombre d'éléments des tableaux internes

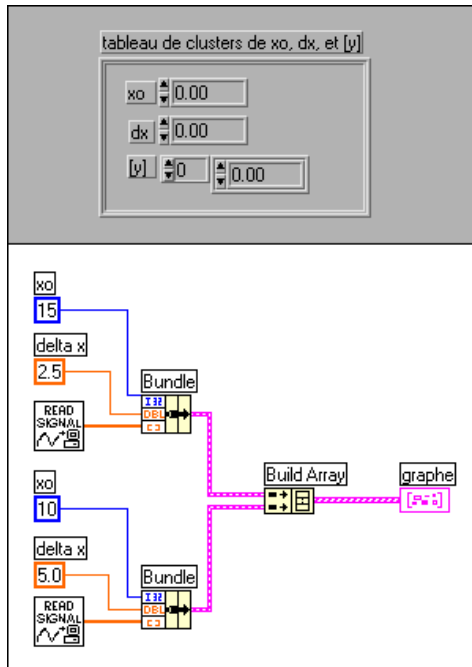
d'un tableau de clusters peut varier. Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



Remarquez que les tableaux sont assemblés dans des clusters qui utilisent la fonction Assembler. De même, les clusters qui en résultent sont intégrés à un tableau avec la fonction “Construire un tableau”. Sinon, vous pouvez utiliser la fonction “Construire un tableau de clusters” qui crée des tableaux de clusters d’entrées spécifiées.

Le cinquième type de données est un tableau de clusters d’une valeur x , une valeur Δx et un tableau de données y . C’est le type de données du graphe oscilloscopique multicourbes le plus commun car vous pouvez spécifier un point de départ unique et incrémenter l’axe des X de chaque tracé.

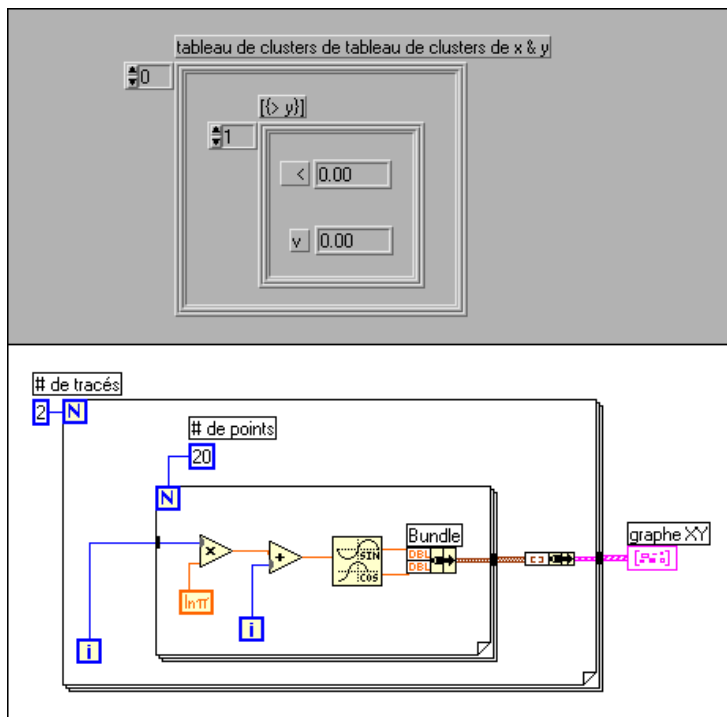
Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



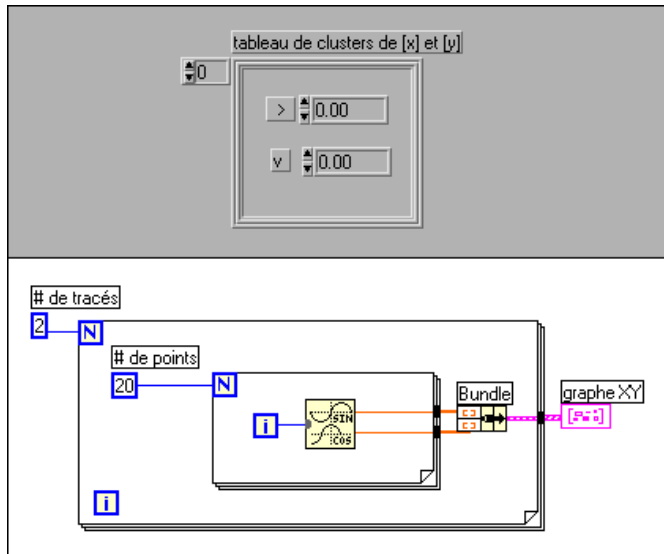
Types de données des graphes XY

Le graphe XY accepte deux types de données multicourbes, décrits dans les paragraphes suivants. Ces deux types de données sont des tableaux de clusters de types de données du tracé unique précédemment décrit.

Le premier type de données est un tableau de clusters de tracés, dans lequel un tracé correspond à un tableau de points. Un point est défini comme un cluster contenant une valeur x et une valeur y . Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



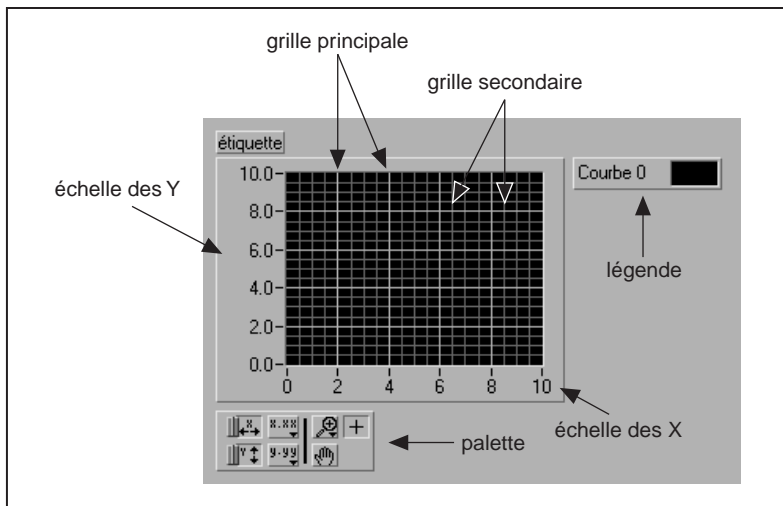
Le deuxième type de données est un tableau de tracés, où un tracé est un cluster d'un tableau x et d'un tableau y . Le diagramme suivant illustre la manière dont vous pouvez créer ce type de données.



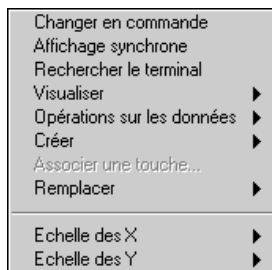
Définir des options personnalisées sur un graphe

Les deux graphes possèdent des options que vous pouvez afficher ou masquer en utilisant le sous-menu **Visualiser** du menu local du graphe. Ces éléments comprennent une légende à partir de laquelle vous pouvez définir la couleur et le style d'un tracé donné, une palette qui sert à modifier les options d'échelle et de format lors de l'exécution d'un VI et une palette **Curseur** pour afficher plusieurs curseurs. Vous trouverez ci-après une illustration d'un graphe qui affiche tous les composants optionnels à

l'exception de la palette **Curseur**, qui est illustrée dans la section *Curseurs du graphe*.



Les graphes possèdent de nombreuses options de personnalisation de l'affichage de vos données. Le menu local du graphe vous est présenté dans l'illustration suivante. **Transposer le tableau** n'est disponible que pour le graphe oscilloscopique.



Options d'échelle

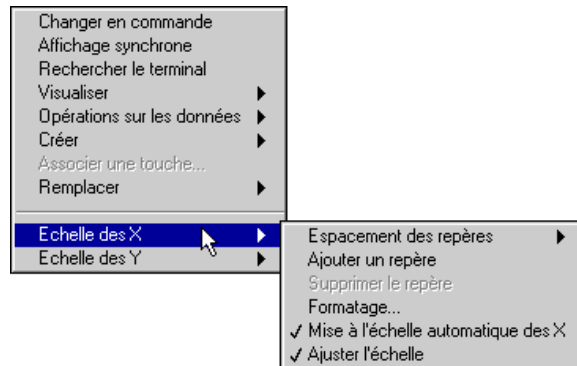
Les graphes peuvent automatiquement ajuster leurs échelles horizontale et verticale afin de représenter au mieux les données qui y sont liées. Activez ou désactivez cette fonction de mise à l'échelle automatique en utilisant les éléments de menu **Mise à l'échelle automatique des X** et **Mise à l'échelle automatique des Y** à partir des sous-menus **Opérations sur les données** ou **Echelle des X/Echelle des Y** du menu local du graphe. Vous pouvez également contrôler ces fonctions de mise à l'échelle automatique par le

biais de la palette du graphe ainsi qu'il est décrit plus loin dans ce chapitre. La mise à l'échelle automatique est activée par défaut. Toutefois, elle peut provoquer un certain ralentissement de l'exécution du graphe.

Vous pouvez directement modifier l'échelle horizontale ou verticale en utilisant l'outil Doigt ou l'outil Texte, exactement comme vous le feriez avec tout autre indicateur ou commande.

Le sous-menu **Opérations sur les données** du menu local du graphe comprend un élément **Rafraîchissement progressif** qui utilise un buffer hors écran afin de réduire le scintillement. Cette fonction peut provoquer un certain ralentissement de l'exécution du graphe en fonction de l'ordinateur et du système vidéo que vous utilisez.

Les échelles des X et des Y ont chacune un sous-menu d'éléments, comme indiqué dans l'illustration suivante.

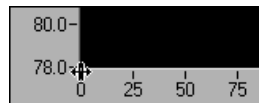


Espace des repères

Par défaut, les valeurs des repères des échelles des x et des y sont réparties de manière uniforme. Si vous le voulez, vous pouvez spécifier n'importe quels emplacements des repères sur les échelles des x ou des y . Cette fonction vous sera utile pour placer des repères de quelques points spécifiques sur un graphe (tels que des points de consigne ou de seuil).

Si vous voulez une répartition des repères non uniforme, choisissez **Echelle des X/Echelle des Y»Espace des repères»Repères arbitraires** dans le menu local de l'échelle. Une fois cette sélection effectuée, si vous déplacez l'outil Doigt sur une marque de graduation, le curseur se transformera en curseur à double flèche présenté dans l'illustration suivante. Vous pouvez à présent créer un nouveau repère en faisant glisser la marque de graduation existante avec la touche <Ctrl> (**Windows**),

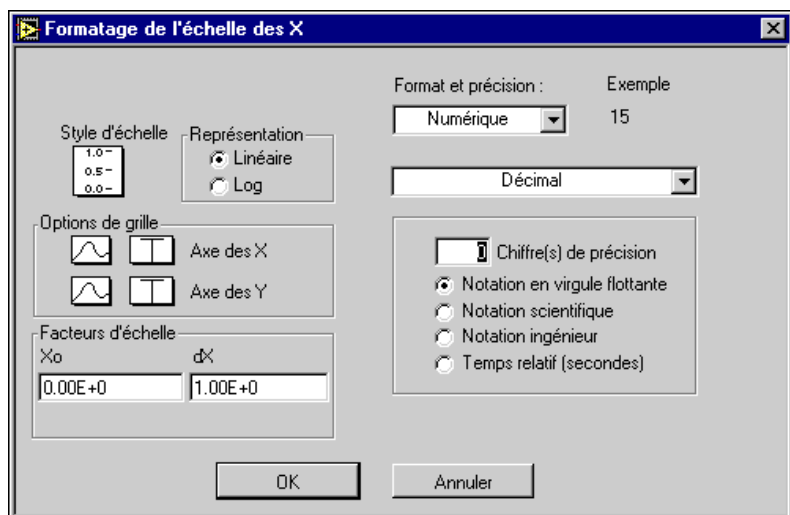
<option> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) enfoncée, ou simplement la déplacer en la faisant glisser.



Vous pouvez ajouter ou supprimer un repère en ouvrant un menu local sur le graphe ou l'échelle et en sélectionnant **Ajouter un repère** ou **Effacer un repère**. Lorsqu'un repère est créé, tapez un nombre dans le repère pour modifier son emplacement.

Formatage

Cliquez sur **Formatage...** pour afficher la boîte de dialogue de l'illustration suivante.



Sélectionnez les éléments dans cette boîte de dialogue afin de définir les propriétés suivantes du graphe.

Style d'échelle : utilisez cet élément pour sélectionner les marques de graduation principales et secondaires de l'échelle. Les marques de graduation principales sont des points qui correspondent aux étiquettes de l'échelle. Les marques de graduation secondaires sont des points internes compris entre les étiquettes. Vous utiliserez également cette palette pour définir si vous voulez que les marques de graduation d'un certain axe soient visibles.

Représentation : utilisez cet élément des menus d'échelle pour définir si l'échelle représente les données en utilisant le mode linéaire ou logarithmique.

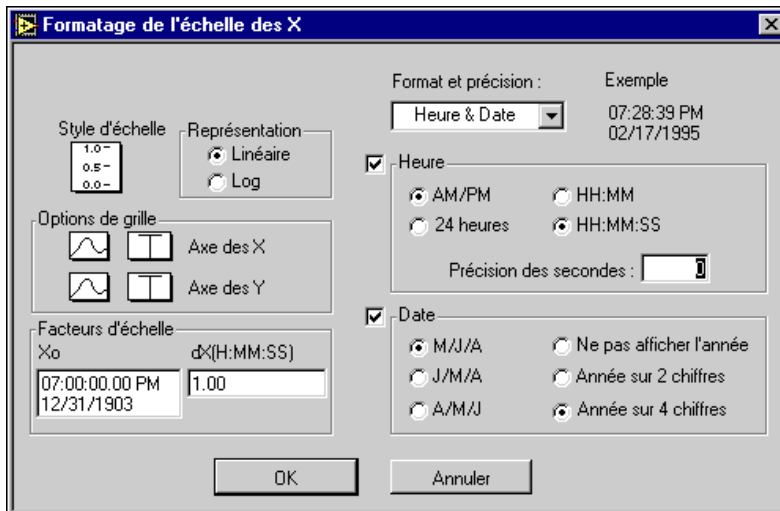
Options de grille : le fait de cliquer sur ces éléments affiche une palette de types de grilles vous permettant de choisir si vous ne voulez pas de quadrillage ou si vous voulez, un quadrillage uniquement aux emplacements des marques de graduation principales (points qui correspondent aux étiquettes de l'échelle) ou aux marques de graduation principales et secondaires (les marques de graduation secondaires sont des points internes compris entre les étiquettes). La commande qui se situe à côté du menu local de la grille est un menu de couleurs que vous pouvez utiliser pour sélectionner la couleur du quadrillage.

Facteurs d'échelle : vous utiliserez les facteurs d'échelle afin de spécifier la valeur initiale et l'espacement entre les points d'un graphe déroulant ou d'un graphe, ou le long des échelles d'un graphe déroulant ou d'un graphe d'intensité. Vous pouvez également utiliser ces facteurs d'échelle pour ajuster l'affichage de vos données. Par exemple, si vos données sont échantillonnées de manière binaire dans une gamme allant de -2 048 à 2 047, ajustez ces données à leurs valeurs de tension appropriée de 0 à 5 en spécifiant un offset supplémentaire de 2,5 et un facteur d'échelle de $5/4095 = 0,001221$.

Format et précision : le format d'échelle des x et des y , en nombres ou en date et heure, peut être sélectionné par le biais d'un menu déroulant situé en haut de la boîte de dialogue.

Si vous sélectionnez un formatage **Numérique**, vous pourrez choisir une notation en virgule flottante, scientifique, ingénieur, ou en temps relatif en secondes. Vous pourrez également choisir une notation décimale, décimale non signée, hexadécimale, octale, ou binaire. Enfin, vous pourrez sélectionner la précision (le nombre de chiffres affichés à la droite du point décimal) de 0 à 20. La précision que vous sélectionnez n'affecte que l'affichage des valeurs. L'exactitude interne dépend toujours de la représentation. Des exemples vous sont fournis dans la boîte de dialogue lors de la sélection de chaque combinaison d'éléments.

Si vous sélectionnez un formatage **Heure et Date**, la boîte de dialogue change comme indiqué dans l'illustration suivante.



Vous pouvez formater soit une heure absolue soit une date absolue, soit les deux. Si vous éditez une échelle et ne tapez que l'heure ou que la date, les composants non spécifiés sont déduits. Si vous n'entrez pas d'heure lors de l'édition, l'heure est automatiquement fixée à minuit. Si vous n'entrez pas de date, la précédente valeur de date est prise en compte. Si vous entrez une date mais que l'échelle n'est pas au format de la date, l'ordre du mois, du jour et de l'année est pris en compte en fonction des paramètres de la boîte de dialogue **Préférences**. Si vous n'entrez que deux chiffres pour l'année, la suite est déduite : tout nombre inférieur à 38 appartient au 21^{ème} siècle, sinon le nombre appartient au 20^{ème} siècle.

Bien que l'heure absolue s'affiche comme une chaîne de caractères d'heure et de date, elle est représentée de manière interne comme le nombre de secondes écoulées depuis minuit le 1^{er} janvier 1904, Temps universel coordonné (UTC), anciennement connu sous le nom de Temps moyen de Greenwich (GMT).

Remarquez les exemples en haut à droite de la boîte de dialogue qui changent au fur et à mesure de vos sélections.

La plage de validité de la date et de l'heure diffère en fonction des plateformes informatiques de la manière suivante :

- **(Windows)** minuit le 2 janvier 1970 – minuit le 3 janvier 2040
- **(Windows NT)** minuit le 1^{er} janvier 1904 – minuit le 3 janvier 2040
- **(Macintosh)** minuit le 2 janvier 1904 – minuit le 2 janvier 2040
- **(UNIX)** minuit le 1^{er} janvier 1904 – minuit le 17 janvier 2038

Ces plages peuvent différer de quelques jours en fonction de votre fuseau horaire et de l'heure d'été en vigueur.

Mise à l'échelle automatique



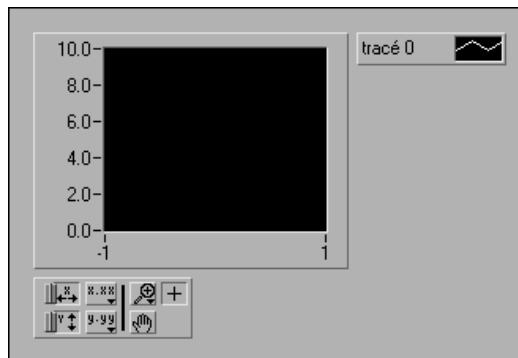
Sélectionnez **Mise à l'échelle automatique (X ou Y)** pour activer ou désactiver cette fonction.

Ajuster l'échelle

Lorsque vous activez l'option **Ajuster l'échelle**, les marques de graduation finales sont arrondies à un multiple de l'incrément utilisé pour l'échelle. Si vous voulez que les échelles soient définies exactement dans la gamme des données, désactivez l'élément **Ajuster l'échelle** du menu local du graphe.

Options de défilement et de zoom

Une palette de **Graphe** est incluse avec n'importe quel graphe déposé sur la face-avant. Cette palette possède des commandes pour le défilement (défilement de la zone d'affichage d'un graphe) et le zoom avant et arrière des sections du graphe. Un graphe accompagné de sa palette vous est présenté dans l'illustration suivante.





Si vous appuyez sur le bouton de **Mise à l'échelle automatique des X**, illustré à gauche, le graphe met automatiquement à l'échelle l'axe des x . Si vous appuyez sur le bouton de mise à l'échelle automatique des y , illustré à gauche, le graphe met automatiquement à l'échelle l'axe des y . Si vous voulez que le graphe mette automatiquement à l'échelle une des échelles de manière continue, cliquez sur le verrou illustré à gauche.



En utilisant les boutons **Format d'échelle**, représentés à gauche, vous pouvez respectivement contrôler le format des marques de graduation des échelles des x et des y pendant l'exécution.

Vous pouvez utiliser les trois autres boutons pour contrôler le mode d'opération du graphe.



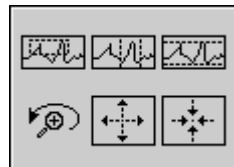
Normalement, vous êtes en mode d'opération standard, indiqué par le plus ou les hachures. En mode d'exécution, vous pouvez cliquer dans le graphe pour déplacer les curseurs.



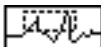
Si vous appuyez sur l'outil Main, représenté à gauche, vous basculez dans un mode dans lequel vous pouvez faire défiler les données visibles en cliquant et en faisant glisser la zone du tracé du graphe.



Si vous appuyez sur l'outil Zoom, illustré à gauche, vous pouvez effectuer un zoom avant ou arrière sur le graphe. Si vous cliquez sur l'outil Zoom, vous verrez un menu local dans lequel vous pourrez choisir parmi les différentes méthodes de zoom. Ce menu vous est présenté dans l'illustration suivante.



Une description de chacun de ces éléments vous est fournie ci-après.



Zoom par rectangle.



Zoom par rectangle, avec un zoom restreint aux données x (l'échelle des y reste inchangée).



Zoom par rectangle, avec un zoom restreint aux données y (l'échelle des x reste inchangée).



Annuler le dernier zoom. Réinitialise le graphe à sa configuration précédente.



Zoom avant sur un point. Si vous maintenez le bouton de la souris appuyé sur un point spécifique, le graphe effectue un zoom avant continu jusqu'à ce que vous relâchiez le bouton.



Zoom arrière sur un point. Si vous maintenez le bouton de la souris appuyé sur un point spécifique, le graphe effectue un zoom arrière continu jusqu'à ce que vous relâchiez le bouton.

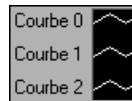


Remarque

A propos des deux derniers modes, zoom avant et zoom arrière sur un point, le fait de maintenir enfoncée la touche <Maj> lors du clic effectue des zooms dans le sens inverse.

Options de légende

Le graphe utilise un style par défaut pour chaque nouveau tracé à moins que vous ne lui attribuez un style de tracé personnalisé. Si vous voulez qu'un graphe multicourbe utilise certaines caractéristiques pour des tracés spécifiques (par exemple, pour que le troisième tracé soit bleu), vous pouvez définir ces caractéristiques en utilisant la légende. Cette dernière peut être affichée ou masquée en utilisant le sous-menu **Visualiser** du menu local du graphe. Vous pouvez également attribuer un nom à chaque tracé en utilisant la légende.



Lorsque vous sélectionnez la **Légende**, un seul tracé s'affiche. Vous pouvez créer plusieurs tracés en faisant glisser un coin de la légende vers le bas à l'aide de l'outil Coin. Une fois que vous avez défini les caractéristiques de la légende, le graphe attribue ces caractéristiques au tracé, sans tenir compte de la visibilité de la légende. Si le graphe reçoit plus de tracés qu'il n'y en a dans la légende, il les dessine avec le style par défaut.

Lorsque vous déplacez le corps du graphe, vous déplacez également la légende. Vous pouvez modifier la position de la légende par rapport au graphe en la faisant glisser vers un nouvel emplacement. Redimensionnez la légende vers la gauche afin de libérer de l'espace pour les étiquettes ou vers la droite pour les échantillons du tracé.

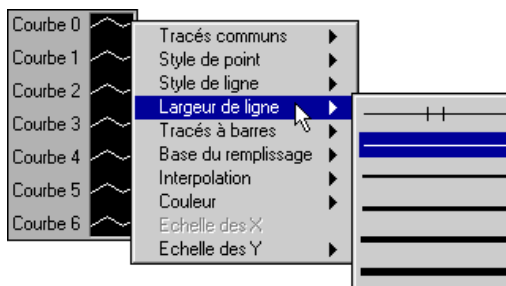
Par défaut, chaque tracé est étiqueté avec un nombre commençant par zéro. Vous pouvez modifier cette étiquette de la même manière qu'une étiquette standard. A la droite de l'étiquette du tracé se trouve l'*échantillon du tracé*. Chaque échantillon de tracé possède son propre menu local qui permet de modifier le tracé, la ligne, la couleur, et les styles de point du tracé. Le tableau de points que vous liez au graphe s'affiche selon les caractéristiques que vous lui avez attribuées dans la légende du graphe.

Le menu local de l'échantillon tracé vous est présenté dans l'illustration suivante.



L'élément **Tracés communs** vous aide à configurer un tracé en un des six styles les plus fréquents comprenant les nuages de points, le tracé à barres et un tracé à remplissage. Les éléments de cette sous-palette sont pré-configurés en point, ligne et style de remplissage à une étape.

Les éléments **Style de point**, **Style de ligne** et **Épaisseur de ligne** affichent des styles que vous pouvez utiliser pour distinguer différents tracés. La sous-palette d'épaisseur de ligne contient des épaisseurs supérieures à la valeur par défaut d'un pixel, ainsi que l'élément filet. Ce dernier élément n'a pas d'effet à l'écran, mais il imprime une ligne très fine si l'imprimante et le mode d'impression supportent l'impression de filet.



 **Remarque**

Sous Windows, les tracés épais larges ne peuvent être qu'en style plein.

L'élément **Tracés à barres** possède une sélection de barres verticales, horizontales ou pas de barre du tout.

L'élément **Base du remplissage** définit jusqu'à quel point la base se remplit. **Zéro** remplit à partir de votre tracé jusqu'à la base générée à 0. **Infini** remplit à partir de votre tracé jusqu'au bord positif du graphe. **Infini** remplit à partir de votre tracé jusqu'au bord négatif du graphe. En utilisant la partie inférieure de ce menu, vous pouvez sélectionner un remplissage jusqu'à un tracé spécifique de ce graphe.

L'élément **Interpolation** affiche la palette présentée dans l'illustration suivante dans laquelle vous pouvez choisir la manière dont le graphe dessine les lignes qui séparent les points d'un tracé. Le premier élément ne dessine pas de ligne, ce qui le rend approprié au nuage de points. L'élément en bas à gauche dessine une ligne droite entre les points tracés. Les deux autres éléments, qui lient les points par un coude à angle droit, sont utiles pour la création des tracés du style histogramme. L'élément en haut à droite effectue le tracé dans la direction des y en premier, et l'élément du bas dans la direction des x en premier.



L'élément **Couleur** affiche la palette de sélection de la couleur du tracé. Vous pouvez également colorer les tracés de la légende avec l'outil Pinceau et modifier les couleurs du tracé lors de l'exécution du VI.

L'élément **Echelle des Y** affiche une liste d'échelles des y sur le graphe. Cette fonction se révèle utile pour des graphes empilés afin de définir à quelle échelle appartient chaque tracé.

Graphes déroulants

Le graphe déroulant est un type spécial d'indicateur numérique qui affiche un ou plusieurs tracés. Les graphes déroulants sont différents des graphes oscilloscopiques dans la mesure où ils conservent les anciennes données jusqu'à une certaine limite que vous pouvez définir. Les nouvelles données sont ajoutées aux anciennes et vous pouvez ainsi voir la valeur actuelle dans son contexte, c'est-à-dire en la comparant avec les anciennes.

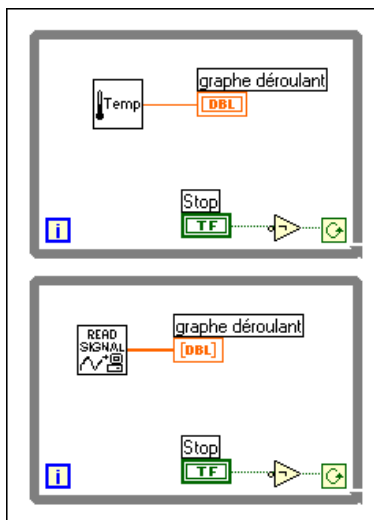
Pour obtenir un exemple de graphe déroulant, consultez la bibliothèque `examples\general\graphs\charts.llb`.

Types de données des graphes déroulants

Vous pouvez transmettre une ou plusieurs valeurs aux graphes déroulants. Comme avec le graphe oscilloscopique, chaque valeur est gérée comme faisant partie d'une forme d'onde uniformément espacée, avec chaque point espacé d'un point du précédent le long de l'axe des X.

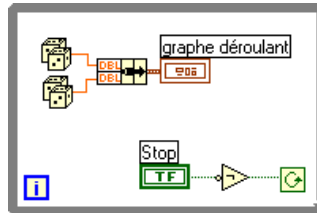
Vous pouvez transmettre une simple valeur scalaire ou un tableau de plusieurs valeurs à un graphe déroulant. Ce dernier gère ces entrées comme de nouvelles données d'un seul tracé.

Vous trouverez ci-après des diagrammes illustrant l'utilisation du graphe déroulant pour chacun de ces types de données.

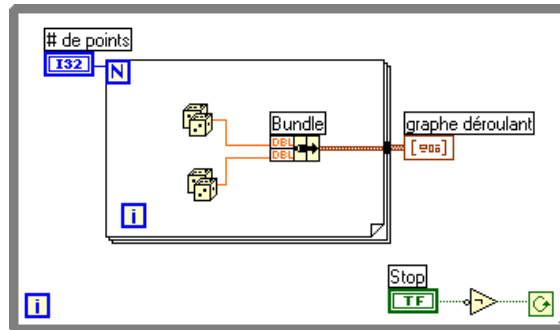


Le graphe déroulant est mis à jour moins fréquemment lorsque vous lui transmettez plusieurs points à la fois plutôt qu'un point à la fois.

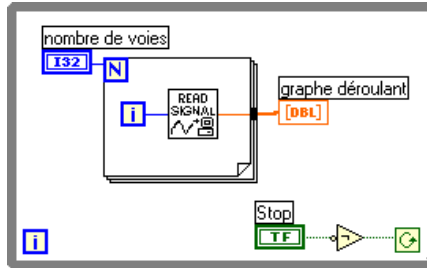
Vous pouvez transmettre de différentes manières des données pour plusieurs tracés dans un même graphe déroulant. La première consiste à assembler les données dans un cluster de nombres scalaires, où chaque nombre représente un point unique pour chacun des tracés. Un exemple vous est proposé dans l'illustration suivante.



Si vous voulez passer plusieurs points de tracés dans une mise à jour unique, vous pouvez alors lier un tableau de clusters de nombres au graphe déroulant. Chaque nombre représente un point unique pour chacun des tracés. L'illustration suivante vous en propose un exemple.

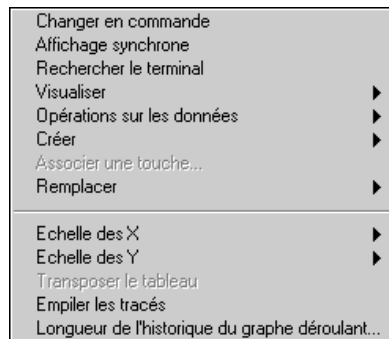


Si le nombre de tracés que vous voulez afficher ne peut être déterminé avant l'exécution, ou si vous voulez passer plusieurs points de tracés en une seule mise à jour, vous pouvez lier un tableau de données à deux dimensions au graphe déroulant. Comme pour le graphe oscilloscopique, les lignes sont normalement gérées comme de nouvelles données pour chaque tracé. Vous pouvez utiliser l'élément **Transposer le tableau** du menu local du graphe déroulant afin de gérer les colonnes comme de nouvelles données pour chaque tracé.



Options des graphes déroulants

Le graphe déroulant possède la plupart des fonctions du graphe oscilloscopique, y compris la légende et la palette, et les deux fonctionnent de manière identique (consultez les sections *Options d'échelle* et *Options de légende* de ce chapitre pour plus d'informations). Le graphe déroulant ne supporte pas les curseurs. L'illustration suivante présente le menu local du graphe déroulant.



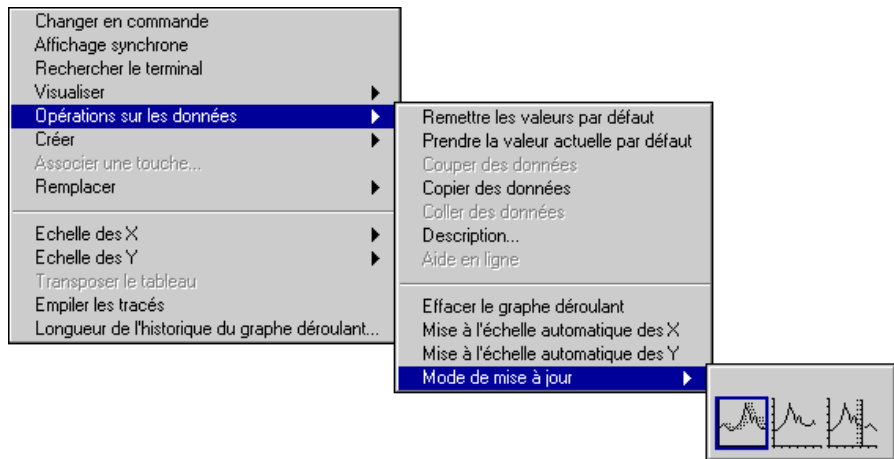
A l'aide du sous-menu **Visualiser** du menu local du graphe déroulant, vous pouvez afficher ou masquer le ou les afficheurs numériques optionnels ou la barre de défilement. L'élément **Afficheur numérique** affiche la dernière valeur tracée. La dernière valeur transmise au graphe déroulant à partir du diagramme est la dernière nouvelle valeur du graphe déroulant. Il y a un afficheur numérique par tracé.

Vous pouvez visualiser les valeurs anciennes contenues dans le buffer en faisant défiler l'axe des x vers une plage de valeurs précédemment tracées en utilisant la barre de défilement ou en modifiant la gamme de l'échelle des x afin de visualiser les anciennes données.

La quantité de données pouvant être conservées par le graphe déroulant dans le buffer est limitée afin d'éviter le manque d'espace mémoire. La taille par défaut de ce buffer est de 1 024 points. Lorsque le graphe déroulant atteint cette limite, les plus anciens points sont supprimés afin de pouvoir accueillir les nouvelles données. Vous pouvez modifier la longueur de ce buffer en utilisant l'élément **Longueur de l'historique du graphe déroulant...** dans le menu local du graphe déroulant.

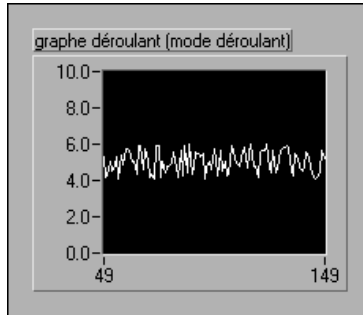
Modes de mise à jour des graphes déroulants

Pour modifier le comportement du graphe déroulant lorsque de nouvelles données sont ajoutées à l'affichage, utilisez l'élément **Mode de mise à jour** du sous-menu **Opérations sur les données** du menu local du graphe déroulant, indiqué dans l'illustration suivante.

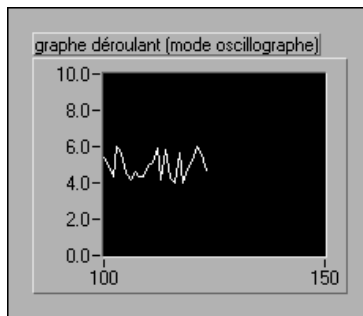


Les trois éléments — graphe déroulant, oscillographe et graphe à balayage — sont illustrés dans les figures suivantes et décrits dans les paragraphes ci-après. Le mode par défaut est le graphe déroulant.

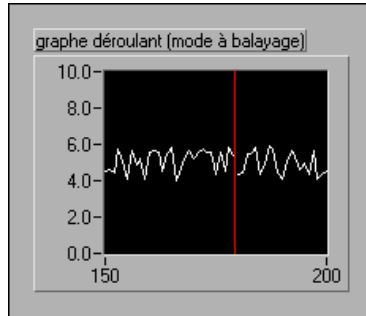
Le mode *graphe déroulant* a un affichage qui défile de manière identique à un enregistreur de graphe déroulant sur papier. Chaque nouvelle valeur reçue est tracée dans la marge de droite et les anciennes valeurs sont décalées vers la gauche.



Le mode *oscillographe* retrace l'affichage comme un oscilloscope. Chaque nouvelle valeur reçue est tracée à droite de la dernière valeur. Lorsque le tracé atteint le bord droit de la zone de traçage, le tracé est effacé et le traçage recommence à partir du bord gauche. L'oscillographe est beaucoup plus rapide que le graphe déroulant car il n'a pas de temps machine de traitement consacré au défilement.

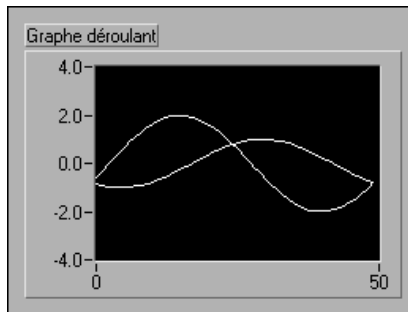


Le mode *graphe à balayage* agit comme un oscillographe mais ne s'efface pas lorsque les données atteignent le bord droit. Au lieu de cela, une ligne verticale mobile marque le début des nouvelles données et se déplace dans l'affichage au fur et à mesure que de nouvelles données sont ajoutées.

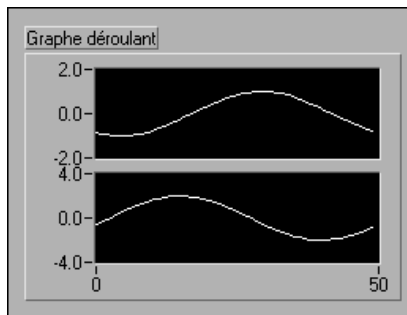


Tracés empilés et tracés superposés

Par défaut, le graphe déroulant affiche plusieurs tracés en les superposant comme des graphes dessinés sur la même grille. Un exemple de tracés superposés vous est fourni dans l'illustration suivante.



Sinon, vous pouvez afficher plusieurs tracés empilés les uns sur les autres avec une échelle des y différente pour chaque tracé. Pour cela, sélectionnez l'élément **Empiler les tracés** du menu local du graphe déroulant. En effectuant cette opération, vous pouvez attribuer une gamme d'échelle des y différente à chaque graphe déroulant. Un exemple de tracés empilés vous est présenté dans l'illustration suivante.

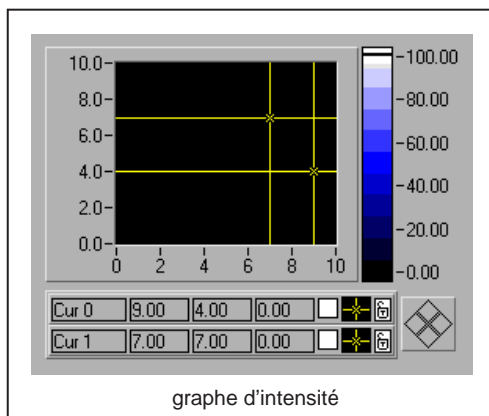
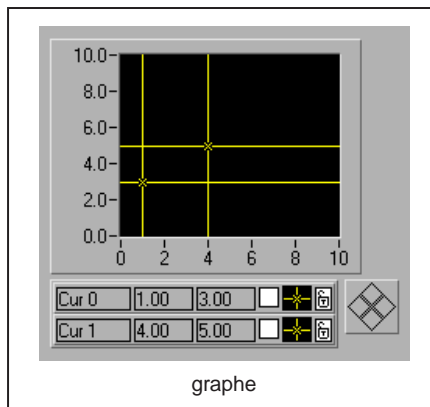


Lorsque vous entrez des données de type cluster dans un graphe déroulant, il empile automatiquement le nombre exact d'affichages de tracés. Lorsque vous entrez des données de type tableau 2D, vous devez créer le nombre exact d'affichages de tracés en utilisant la légende par le biais du sous-menu **Visualiser** du menu local. Le nombre d'affichages de tracés empilés augmente en fonction de la taille de l'affichage de la légende.

Curseurs du graphe

Pour chaque graphe, vous pouvez visualiser une palette **Curseur** utilisée pour mettre des curseurs sur le graphe. Vous pouvez étiqueter le curseur sur le tracé et utiliser le curseur comme marque de graduation. Lorsque vous utilisez un curseur comme marque de graduation, vous pouvez le verrouiller à un tracé de données de manière à ce que le curseur suive les données.

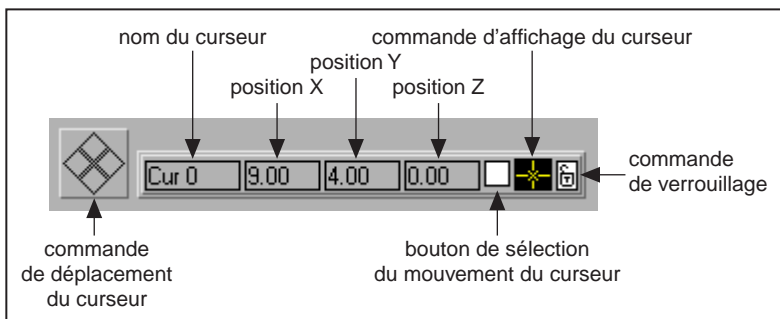
Vous trouverez ci-après des illustrations d'un graphe oscilloscopique et d'un graphe d'intensité avec la palette **Curseur** affichée.



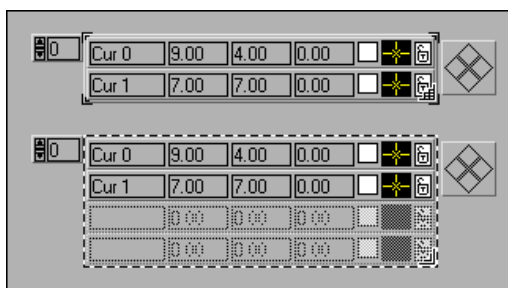
Chaque curseur de graphe comporte les éléments suivants.

- Une étiquette
- Des coordonnées x et y et des coordonnées z le cas échéant
- Un bouton qui permet de déplacer le curseur sur le tracé
- Un bouton qui contrôle l'aspect du curseur
- Un bouton qui détermine si le curseur est verrouillé à un tracé ou s'il peut être librement déplacé.

Ces éléments vous sont présentés dans l'illustration suivante.



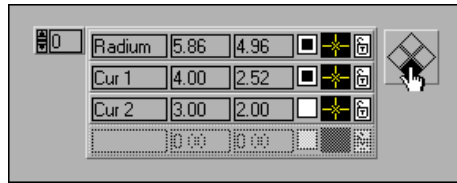
La palette **Curseur** fonctionne comme un tableau. Vous pouvez l'étirer pour afficher des informations sur de nombreux curseurs et utiliser la commande d'indice pour visualiser d'autres informations de curseur dans la palette. Utilisez l'élément **Visualiser** du menu local de l'affichage du nom du curseur pour visualiser la commande d'indice.



Pour supprimer un curseur, vous devez le sélectionner en utilisant les éléments **Commencer la sélection** et **Terminer la sélection** du menu local **Opérations sur les données** et coupez ensuite le curseur à l'aide de l'élément **Couper** du même menu. Pour plus d'informations, consultez la

section *Sélection des cellules du tableau* du chapitre 14, *Commandes et indicateurs de tableau et de cluster*.

Vous pouvez déplacer un curseur sur le graphe en le faisant glisser à l'aide de l'outil Doigt ou en utilisant la commande de déplacement du curseur. Pour faire glisser le curseur, vérifiez que les outils Main ou Zoom ne sont pas sélectionnés dans le graphe. En cliquant sur les flèches de la commande de déplacement du curseur, vous pouvez déplacer tous les curseurs sélectionnés dans la direction correspondante. Vous sélectionnez les curseurs soit en les déplaçant sur le graphe avec l'outil Doigt soit en cliquant sur le bouton de sélection de déplacement de curseur. Dans l'exemple suivant, les deux curseurs du haut se déplacent verticalement vers le bas.



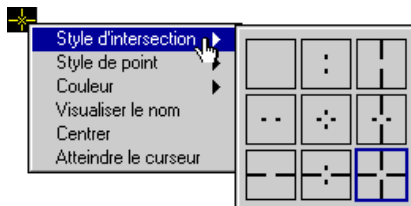
Commande d'affichage du curseur



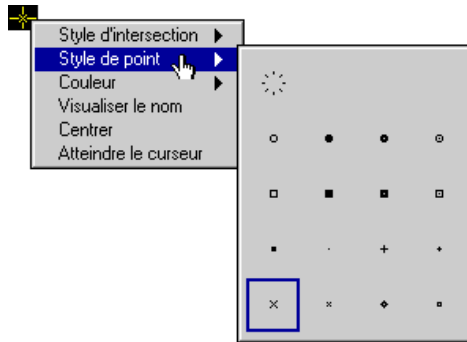
En cliquant sur la commande d'affichage du curseur avec l'outil Doigt, vous pouvez afficher un menu local qui sert à contrôler l'aspect du curseur et la visibilité du nom du curseur sur le tracé. Ce menu local vous est présenté dans l'illustration suivante.



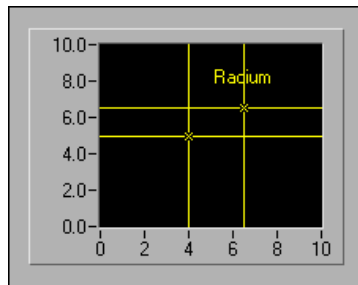
Ce menu vous permet de sélectionner le style d'intersection. L'intersection peut prendre la forme d'une ligne verticale et/ou horizontale extensible à l'infini, d'une ligne verticale et/ou horizontale de plus petite taille centrée sur l'emplacement du curseur ou une forme invisible, comme indiqué dans l'illustration suivante.



Vous pouvez également choisir le style de point à utiliser pour le marquage de l'emplacement du curseur et la couleur du curseur comme indiqué dans l'illustration suivante.



Sélectionnez l'élément **Visualiser le nom** dans ce menu pour faire en sorte que le nom du curseur soit visible sur le tracé, comme indiqué dans l'illustration suivante.



Le fait de sélectionner **Centrer** place le curseur dans la zone d'affichage du graphe. Cette fonction est utile lorsque le curseur n'est plus visible. La sélection de cet élément modifie la position des coordonnées (x,y) du curseur.

Si vous choisissez **Atteindre le curseur**, vous déplacez la zone d'affichage du graphe de manière à ce que le curseur soit visible. La position du curseur ne change pas mais les échelles sont modifiées afin d'afficher le curseur sélectionné. La taille de la zone d'affichage ne change pas non plus. Cette fonction est utile lorsque le curseur est utilisé pour marquer un point d'intérêt du graphe, tel que le minimum ou le maximum, et que vous voulez visualiser ce point.

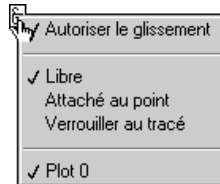


déverrouillé



verrouillé

Vous pouvez utiliser le dernier bouton de chaque curseur de manière à le verrouiller sur un tracé donné. En cliquant sur le bouton de verrouillage, vous faites apparaître un menu local que vous pouvez utiliser pour verrouiller le curseur à un tracé spécifique. Si vous verrouillez le curseur à un tracé, le bouton affiche un cadenas fermé. Ce menu local vous est présenté dans l'illustration suivante.



L'élément **Glisser** détermine si vous pouvez déplacer le curseur avec la souris. Si l'option **Autoriser le glissement** est sélectionnée, vous pouvez déplacer ou faire glisser le curseur. Les éléments qui se situent sous la ligne pointillée du menu déterminent la manière dont vous pouvez déplacer le curseur avec la souris. Si l'option **Autoriser le glissement** est sélectionnée, vous pouvez déplacer le curseur sur le tracé.

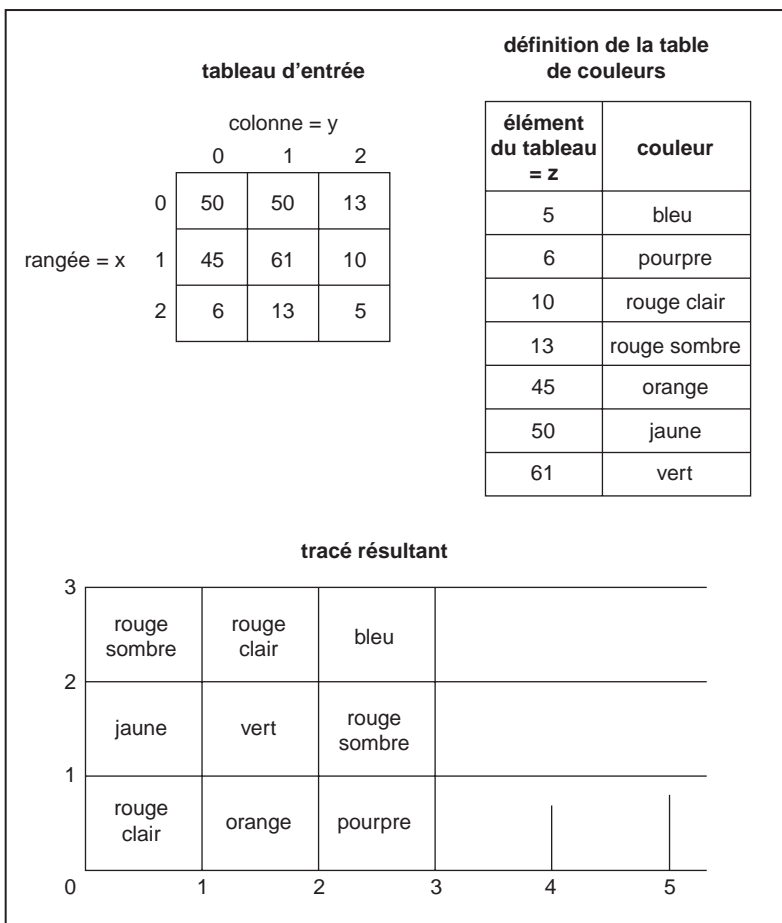
Sélectionnez **Libre** si vous voulez placer ou déplacer le curseur à n'importe quel endroit du graphe. Sélectionnez **Attaché au point** si vous voulez que le curseur soit toujours attaché au point le plus proche d'un tracé. Sélectionnez **Verrouiller au tracé** pour attacher le curseur à un tracé spécifique. La première fois que vous sélectionnez **Verrouiller au tracé**, le curseur s'attache au premier point du tracé. Une fois que vous avez libéré le curseur et que vous l'avez déplacé sur une nouvelle position, le fait de sélectionner **Verrouiller au tracé** déplace le curseur au dernier emplacement du curseur verrouillé.

Sous la deuxième ligne pointillée de ce menu se trouve une liste des tracés auxquels vous pouvez verrouiller le curseur (par exemple, Tracé 0, Tracé 1, Tracé 2, etc.).

Un grand nombre d'éléments sont disponibles pour la création, le contrôle et la lecture par programme des curseurs ou des marques de graduation. Pour y accéder, utilisez l'attribut `node` des graphes. Pour plus d'informations, consultez le chapitre 22, *Attribute Nodes*.

Graphes déroulants d'intensité

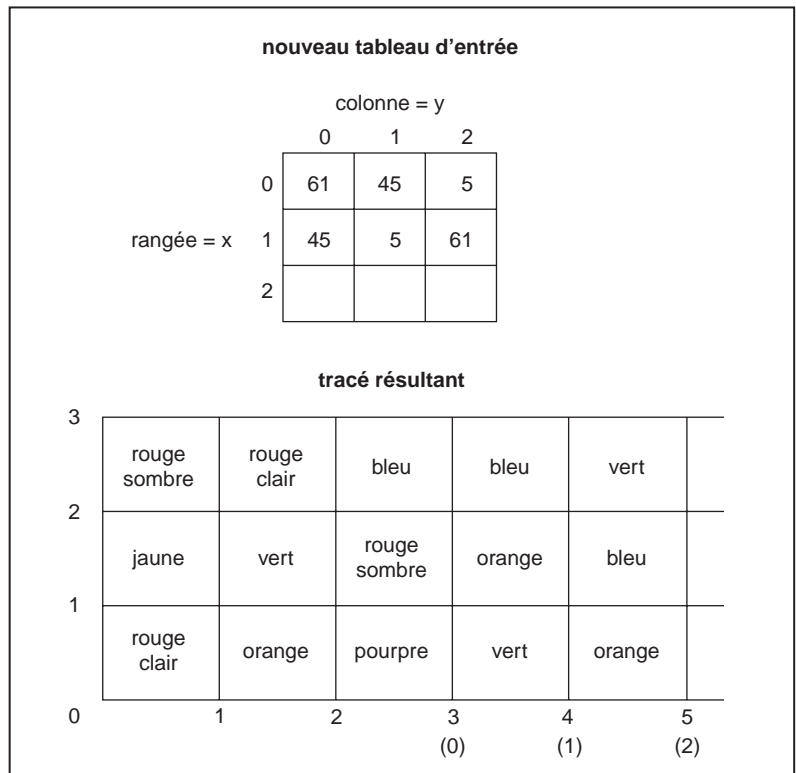
Le graphe déroulant d'intensité constitue une manière d'afficher trois dimensions de données sur un tracé à deux dimensions en plaçant des blocs de couleurs sur un plan cartésien. Le graphe déroulant d'intensité accepte un tableau de nombres à deux dimensions. Chaque nombre du tableau représente une couleur spécifique. Les indices d'un élément du tableau à deux dimensions définissent l'emplacement de ce tracé pour cette couleur. L'illustration suivante montre le concept de l'utilisation du graphe déroulant d'intensité.



Vous pouvez définir les couleurs du graphe déroulant d'intensité de manière interactive en utilisant l'échelle de couleurs. Ou alors, vous pouvez définir les couleurs par programme par le biais d'un attribut node du graphe déroulant. La section *Représentation des couleurs* plus loin dans ce chapitre explique la procédure d'attribution d'une couleur à un nombre.

Les indices du tableau correspondent au coin inférieur gauche du bloc de couleur. Le bloc possède une zone d'unités définie selon les indices du tableau. Le graphe déroulant d'intensité peut afficher jusqu'à 256 couleurs distinctes.

Une fois un bloc de données tracé, l'origine du plan cartésien se décale à droite du dernier bloc de données. Lorsque de nouvelles données sont envoyées au graphe déroulant d'intensité, elles s'affichent à droite des anciennes données, comme indiqué dans l'illustration suivante.



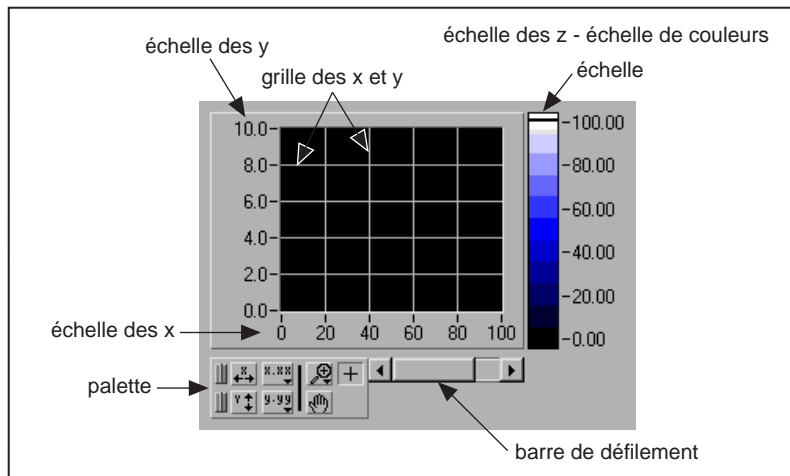
Lorsque l’affichage du graphe déroulant est plein, les données les plus anciennes sont décalées vers la gauche en dehors du graphe déroulant.

Consultez les exemples de graphes déroulants d’intensité dans la bibliothèque `exemples\general\graphs\intgraph.llb`.

Options des graphes déroulants d’intensité

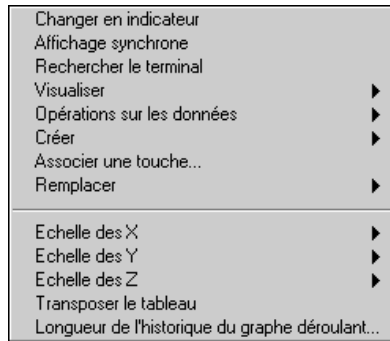
Le graphe déroulant d’intensité partage un grand nombre d’éléments optionnels avec les autres graphes déroulants. La plupart d’entre eux peuvent être affichés ou masqués par le biais du sous-menu **Visualiser** du menu local du graphe. Ces éléments comprennent une palette que vous pouvez utiliser pour modifier l’échelle et le format lors de l’exécution du VI. De plus, étant donné que le graphe déroulant d’intensité possède une troisième dimension (la couleur), une échelle similaire à une commande de rampe de couleurs définit la gamme et la représentation des valeurs en couleurs.

Vous trouverez ci-après un graphe déroulant montrant l’ensemble des composants optionnels.



Les graphes déroulants d’intensité ont un certain nombre d’éléments que vous pouvez utiliser afin de personnaliser l’affichage de vos données.

Le menu local du graphe déroulant d'intensité vous est présenté dans l'illustration suivante.



La plupart de ces éléments sont identiques aux éléments de menu du graphe déroulant. Grâce au menu **Visualiser**, vous pouvez afficher et masquer l'échelle des couleurs pour l'échelle des z. Les éléments Echelle des X et Echelle des Y sont identiques aux menus correspondants du graphe déroulant.

Le graphe déroulant d'intensité conserve un historique des données des précédentes mises à jour. Vous pouvez configurer ce buffer en sélectionnant **Longueur de l'historique du graphe déroulant...** dans le menu local du graphe déroulant. La taille par défaut pour un graphe déroulant d'intensité est de 128 points. Remarquez que l'affichage du graphe déroulant d'intensité peut être gourmand en mémoire. Par exemple, l'affichage d'un graphe déroulant simple précision avec un historique de 512 points et 128 valeurs y nécessite $512 * 128 * 4$ octets (taille d'un simple), ou 256 Kilo-octets. Si vous voulez afficher de grandes quantités de données sur un graphe déroulant d'intensité, vérifiez que la mémoire disponible est suffisante.

Le graphe déroulant d'intensité supporte les modes de mise à jour standard du graphe déroulant, du graphe à balayage et de l'oscillographe. Comme avec le graphe déroulant, vous pouvez sélectionner le mode de mise à jour dans le menu **Opérations sur les données**.

Représentation des couleurs

Vous pouvez définir la représentation des couleurs de manière interactive comme vous définissez les couleurs d'une commande numérique de rampe de couleurs. Pour plus de détails, consultez la section *Rampe de couleurs* dans le chapitre 9, *Commandes et indicateurs numériques*.

Par le biais d'un attribut node, il existe deux manières de définir des couleurs par programme. La première méthode consiste à spécifier la représentation valeur-couleur à l'attribut node de manière identique à l'échelle des couleurs. Pour cette méthode, vous devez spécifier l'attribut **Info de l'échelle des Z : Tableau de couleurs**. Cet attribut consiste en un tableau de clusters, dans lequel chaque cluster contient une valeur numérique limite ainsi que la couleur correspondante à afficher pour cette valeur. Lorsque vous spécifiez la table des couleurs de cette manière, vous pouvez également spécifier une couleur supérieure hors gamme en utilisant l'attribut **Info de l'échelle des Z : Couleur supérieure**, une couleur inférieure hors gamme en utilisant **Info de l'échelle des Z : Couleur inférieure**. Le nombre total de couleurs est limité à 254 couleurs, plus la couleur inférieure et la couleur supérieure hors gamme qui ramènent le total à 256 couleurs. Si vous spécifiez plus de couleurs, la table est créée par l'interpolation des couleurs spécifiées.

La deuxième manière de définir des couleurs par programme consiste à spécifier une table de couleurs en utilisant l'attribut **Table des couleurs**. Avec cette méthode, vous pouvez spécifier un tableau qui peut atteindre 256 couleurs. Les données transmises au graphe déroulant sont représentées en indices dans cette table des couleurs en tenant compte de l'échelle des couleurs. Si la gamme de l'échelle des couleurs va de zéro à 100, une valeur de zéro dans les données est représentée en indice et une valeur de 100 est représentée en indice 254, avec des valeurs internes interpolées entre un et 254. Tout ce qui se trouve en-dessous de zéro est représenté dans la couleur hors gamme par défaut (indice zéro) et tout ce qui dépasse 100 est représenté dans la couleur hors gamme par excès (indice 255).



Remarque

Les couleurs affichées par votre graphe déroulant d'intensité (ou graphe) sont limitées aux couleurs exactes et au nombre de couleurs que votre carte vidéo peut afficher. Vous êtes également limité par le nombre de couleurs allouées à votre affichage.

Graphes d'intensité

Le graphe d'intensité est pratiquement identique au graphe déroulant d'intensité à l'exception du fait qu'il ne conserve pas les précédentes données. A chaque fois qu'il reçoit de nouvelles données, ces dernières remplacent les anciennes à leur arrivée.

Pour voir un exemple de graphe d'intensité, consultez la bibliothèque `examples\graphs\intgraph.llb`.

Type de données des graphes d'intensité

Le graphe d'intensité accepte des tableaux de nombres à deux dimensions, où chaque nombre est représenté dans le graphe déroulant par une couleur.

Les lignes de données que vous transmettez s'affichent comme de nouvelles colonnes sur le graphe déroulant. Si vous voulez que les lignes s'affichent en tant que lignes, utilisez l'élément **Transposer le tableau** dans le menu local du graphe déroulant.

Options des graphes d'intensité

Le graphe d'intensité fonctionne plus ou moins comme le graphe déroulant d'intensité à l'exception du fait qu'il ne possède pas les modes de mise à jour. Etant donné que chaque mise à jour remplace les données précédentes, il ne possède pas de barre de défilement ni d'options d'historique.

Le graphe d'intensité peut avoir des curseurs comme les autres graphes. Chaque curseur affiche les valeurs x , y , et z d'un point spécifié sur le graphe. Pour plus d'informations sur la manipulation des curseurs sur le graphe, consultez la section *Curseurs du graphe* de ce chapitre.

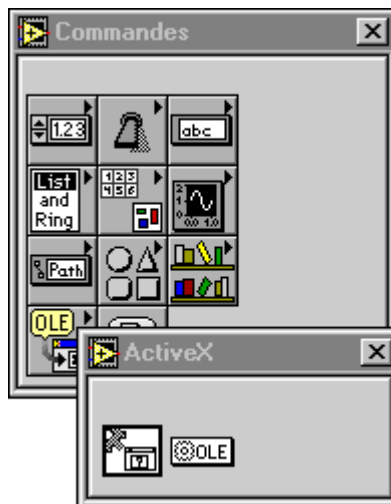
Vous pouvez définir la représentation des couleurs de la même manière qu'avec le graphe déroulant d'intensité.

Commandes ActiveX

Ce chapitre décrit les capacités du container ActiveX. Ce dernier améliore les interactions entre les logiciels G et d'autres applications.

Eléments ActiveX de la face-avant

La face-avant comprend une sous-palette **ActiveX** qui contient les deux commandes ActiveX de l'illustration suivante : le container ActiveX et la variable ActiveX.



Commande et indicateur de variable ActiveX

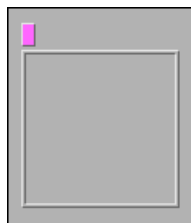
La commande et l'indicateur de variable ActiveX vous permettent de transmettre des données de variable ActiveX au logiciel afin d'améliorer la fonctionnalité du client ActiveX. Lorsque vous placez la commande et l'indicateur de variable ActiveX sur le diagramme, il s'affiche de la manière suivante.



Utilisez cet objet de la face-avant lorsque les données de variable ActiveX sont converties afin d'afficher des données.

Container ActiveX

Le container ActiveX permet l'insertion d'objets ActiveX dans une face-avant de VI avec des commandes intégrées. Vous pouvez utiliser ce container pour afficher sur la face-avant des commandes ActiveX et des documents incorporés. L'illustration suivante présente le container tel qu'il s'affiche lorsque vous le placez pour la première fois sur la face-avant.



Le container ActiveX s'affiche sur le diagramme en tant que terminal de refnum automation. Vous pouvez lier ce terminal aux fonctions Automation et par conséquent contrôler l'objet incorporé dans le container. Si cet objet n'a pas d'interface Automation, le terminal aura un refnum invalide et ne pourra pas être utilisé avec les fonctions Automation.

Pour insérer un objet dans le container de la face-avant, ouvrez un menu local et sélectionnez **Insérer un objet ActiveX**. La boîte de dialogue **Sélectionner un objet ActiveX** s'affiche.

Il existe deux types généraux d'objets qui peuvent être insérés : les documents ActiveX et les commandes ActiveX.

Documents ActiveX : ces objets peuvent être contenus par l'objet container et édités en ouvrant un menu local et en sélectionnant l'option **Modifier l'objet**. Cette action affiche une nouvelle fenêtre de modification. Certains documents supportent l'interface automation et peuvent être utilisés avec les fonctions automation sur le diagramme.

Commandes ActiveX : ces objets peuvent être activés et exécutés à l'intérieur même de l'objet container. Ils ont des interfaces automation et peuvent être contrôlés en utilisant les fonctions Automation sur le diagramme.

Les objets peuvent être déposés de trois manières différentes, que vous pouvez choisir dans le menu déroulant du haut de la boîte de dialogue.

Créer un document : choisissez un type de document parmi ceux qui sont enregistrés dans le système.

Créer un objet à partir d'un fichier : choisissez un des documents situés dans le système de fichiers. L'objet peut soit rester lié au fichier, soit être copié de manière statique dans la face-avant.

Créer une commande : choisissez une commande ActiveX parmi celles qui sont enregistrées sur le système.



Deux types d'objets peuvent être placés dans le container : les commandes et les documents. Vous pouvez créer une nouvelle commande ou un nouveau document, ou insérer un document existant. Pour créer un nouveau document, sélectionnez **Créer un document** (comme indiqué dans la précédente illustration) et choisissez le type d'objet parmi les éléments répertoriés. Pour insérer un document ou un fichier existant, sélectionnez

Créer un objet à partir d'un fichier. La boîte de dialogue est remplacée par celle de l'illustration suivante.



Utilisez le bouton **Parcourir...** pour rechercher un document à insérer. Si vous sélectionnez l'option **Attacher au fichier**, le document se mettra à jour lorsque l'objet de la face-avant sera modifié. Si vous ne sélectionnez pas **Attacher au fichier**, une version statique du document sera insérée.

Pour insérer une commande existante, sélectionnez **Créer une commande**. La boîte de dialogue est remplacée par celle de l'illustration suivante.



Les types de commandes disponibles sont enregistrés avec le système.

Construire des palettes ActiveX

Par le biais de **Projet»Importer des commandes ActiveX...** vous pouvez également convertir un jeu de commandes ActiveX en commandes personnalisées et l'ajouter aux menus de la palette.

Lorsque l'élément de menu est sélectionné, une liste des commandes du système s'affiche. Vous pouvez ensuite sélectionner une ou plusieurs commandes. Vous êtes alors invité à choisir un répertoire ou un fichier `.lib` dans lequel les commandes personnalisées seront stockées. L'emplacement par défaut est `user.lib`, car tous les fichiers et les répertoires de ce répertoire s'affichent automatiquement dans les menus de la palette.

Programmation par diagramme

Cette section contient des informations sur les composants requis pour construire et manipuler un diagramme en G.

La Partie III, *Programmation par diagramme*, contient les chapitres suivants :

- Le chapitre 17, *Introduction au diagramme*, décrit les terminaux et les nœuds - deux des trois éléments que vous utilisez pour construire un diagramme. Le troisième élément, le câblage, est abordé dans le chapitre 18, *Câblage du diagramme*.
- Le chapitre 18, *Câblage du diagramme*, explique comment connecter des terminaux du diagramme en les reliant avec des fils de liaison.
- Le chapitre 19, *Structures*, explique comment utiliser la boucle For, la boucle While, la structure Condition et la structure Séquence. Ces structures se trouvent dans la palette **Fonctions»Structures**.
- Le chapitre 20, *Boîtes de calcul*, explique comment utiliser Formula Nodes pour exécuter des opérations mathématiques sur le diagramme. La boîte de calcul est disponible à partir de la palette **Fonctions»Structures**.
- Le chapitre 21, *VI Serveur*, décrit le mécanisme permettant de contrôler les VIs et les applications au moyen d'un programme. Ce chapitre explique également comment contrôler les VIs ou les applications à distance.

- Le chapitre 22, *Attribute Nodes*, explique comment utiliser les attribute nodes pour définir et lire les attributs des commandes de la face-avant au moyen d'un programme. Parmi les attributs utiles figurent les couleurs d'affichage, la visibilité de la commande, les chaînes de caractères des menus pour une commande à menu déroulant, les couleurs des tracés des graphes ou des graphes déroulants et les curseurs des graphes.
- Le chapitre 23, *Variables globales et locales*, explique comment définir et utiliser des variables globales et des variables locales. Utilisez des variables globales pour accéder facilement à un ensemble de valeurs particulier à partir de plusieurs VIs. Les variables locales remplissent un rôle similaire au sein d'un VI unique.

Introduction au diagramme

Ce chapitre présente les terminaux et les nœuds – deux des trois éléments que vous utilisez pour concevoir un diagramme. Le troisième élément, le câblage, est traité dans le chapitre 18, *Câblage du diagramme*.

Terminaux et nœuds

Les diagrammes sont constitués de terminaux, de nœuds et de fils de liaison.

Les *terminaux* sont des ports par lesquels les données passent entre le diagramme et la face-avant, ainsi qu'entre les nœuds et le diagramme. Ils sont également à la base des icônes de fonctions et de VIs. L'illustration suivante montre un exemple de terminal associé à une icône. Pour afficher les terminaux d'une fonction ou d'un VI, ouvrez un menu local sur l'icône et sélectionnez **Visualiser»Terminaux**.



Les *nœuds* sont des éléments d'exécution de programme. Ils sont analogues aux instructions, opérateurs, fonctions et sous-programmes des langages de programmation conventionnels.

Les *fils de liaison* sont des chemins de données entre les terminaux d'entrée et de sortie.

Terminaux

Le G possède plusieurs types de terminaux. En général, un terminal est tout point auquel vous pouvez attacher un fil. Le G possède des terminaux de commande et d'indicateur, des terminaux de nœud, des *constantes* et des terminaux spécialisés sur les structures.

Les terminaux qui fournissent des données, tels que les terminaux de commande de face-avant, terminaux de sortie de nœud et constantes, sont également appelés des *terminaux source*. Les terminaux d'entrée de nœud et les terminaux d'indicateur de face-avant sont également appelés des

terminaisons de destination ou des terminaux de destination car ils reçoivent les données.

Terminaux de commande et d'indicateur

Vous entrez des valeurs dans les commandes de la face-avant et, lors de l'exécution d'un VI, les terminaux de commande passent ces valeurs au diagramme. A la fin de l'exécution du VI, les données de sortie sont transmises du diagramme aux indicateurs de la face-avant par le biais des terminaux d'indicateur.

Les symboles de certains terminaux de commande et d'indicateur du G vous sont présentés dans le tableau 17-1, *Symboles de terminaux de commande et d'indicateur du G*. Chaque symbole comprend une image et possède une couleur qui indique le type de données de la commande ou de l'indicateur et, dans le cas des nombres, la représentation. Les terminaux de commande ont une bordure plus épaisse que les terminaux d'indicateur. Etant donné que le terminal appartient à sa commande ou à son indicateur correspondant, vous ne pouvez pas supprimer un terminal ; si vous voulez supprimer une commande ou un indicateur, faites-le à partir de la face-avant.

Un terminal de tableau comprend un des types de données indiqués entre crochets et prend la couleur du type de données des éléments du tableau.

Tableau 17-1. Symboles de terminaux de commande et d'indicateur du G









































Commande	Indicateur	Description	Couleur
		Virgule flottante précision étendue	Orange
		Virgule flottante double précision	Orange
		Virgule flottante simple précision	Orange
		Virgule flottante précision étendue complexe	Orange
		Virgule flottante double précision complexe	Orange
		Virgule flottante simple précision complexe	Orange

Tableau 17-1. Symboles de terminaux de commande et d'indicateur du G (Suite)

Commande	Indicateur	Description	Couleur
		Entier 32 bits non signé	Bleu
		Entier 16 bits non signé	Bleu
		Entier 8 bits non signé	Bleu
		Entier 32 bits (mot long)	Bleu
		Entier 16 bits (mot)	Bleu
		Entier 8 bits	Bleu
		Cluster	Marron ou rose
		Tableau	Varié
		Chemin	Cyan
		Refnum	Cyan
		Booléen	Vert
		Chaîne de caractères	Rose
		Énumération	Bleu
		Variant OLE	Pourpre

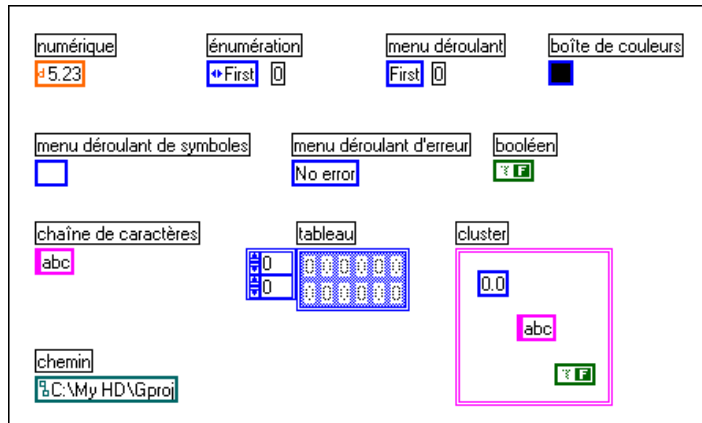
Constantes

Les constantes sont des terminaux sur un diagramme qui lui fournissent directement des valeurs de données. Les *constantes définies par l'utilisateur* sont définies lors de l'édition de la constante avant l'exécution du programme, mais vous ne pouvez pas modifier leurs valeurs durant l'exécution. Les *constantes universelles* ont des valeurs fixes.

Constantes définies par l'utilisateur

La manière la plus facile de créer une constante définie par l'utilisateur consiste à ouvrir un menu local sur une entrée et à sélectionner **Créer une constante**. Ces constantes sont également accessibles à partir de différentes sous-palettes de la palette **Fonctions** selon leur type. La majorité se situe en bas ou en haut de la palette correspondante, telle que la constante numérique, la constante type énuméré et la constante menu déroulant dans la ligne du bas ou dans la palette **Numérique**. Trois constantes — la constante boîte de couleurs, la constante symboles et la constante liste déroulante d'erreurs — se trouvent dans la sous-palette **Numérique**» **Constantes numériques supplémentaires**. La constante chemin se trouve dans la sous-palette **E/S sur fichiers**» **Constantes de fichiers**.

L'illustration suivante vous montre les différentes constantes.

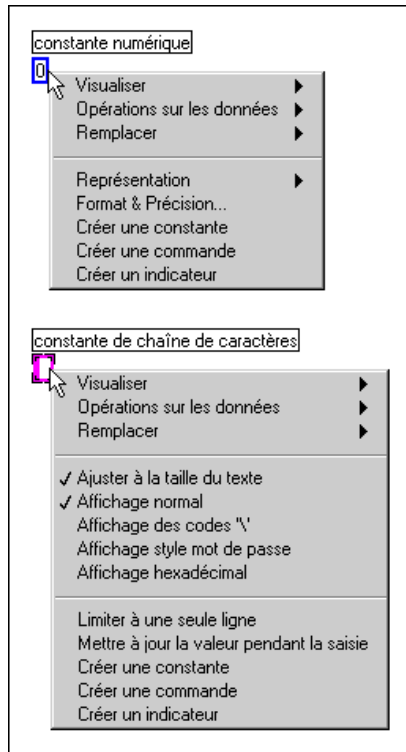


Vous pouvez étiqueter votre constante en ouvrant un menu local sur la constante et en sélectionnant **Visualiser**» **Étiquette**. Une zone de texte en surbrillance s'affiche, ce qui indique que vous pouvez y taper du texte immédiatement sans avoir à changer d'outil.

Les constantes définies par l'utilisateur, telles que les étiquettes, sont automatiquement redimensionnées au fur et à mesure que vous y entrez des informations. Si vous redimensionnez ou modifiez la forme d'une étiquette ou d'une constante chaîne de caractères, vous pouvez sélectionner **Ajuster à la taille du texte** dans son menu local pour que la constante ou l'étiquette soit automatiquement redimensionnée à la taille de son contenu.

Vous pouvez utiliser l'outil Doigt pour définir la valeur d'une constante définie par l'utilisateur exactement comme pour la valeur d'une commande

numérique, un interrupteur à glissière booléen ou une commande de chaîne de caractères sur la face-avant. Les *constantes numériques* et *chaînes de caractères* ressemblent aux commandes numériques de la face-avant et possèdent des menus locaux identiques à ceux des commandes, comme le montre l'illustration suivante.



Vous pouvez utiliser les touches fléchées pour incrémenter ou décrémenter une nouvelle constante numérique ou une constante numérique dont la valeur est sélectionnée. Cela vous sera particulièrement utile avec des paramètres de commande de programmation qui possèdent des valeurs faibles telles que 1, 2 ou 3.

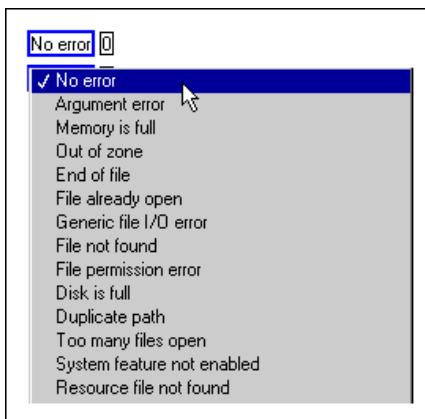
La *constante menu déroulant* associe du texte et des nombres, comme une commande menu déroulant de la face-avant. La valeur de la constante menu déroulant est un entier 16 bits non signé. Bien que vous puissiez modifier la représentation d'une constante menu déroulant en n'importe quel type numérique excepté le type complexe, la valeur reste toujours un entier compris entre zéro et $n - 1$.

La *constante type énumérée* ressemble à la constante menu déroulant excepté le fait que les mnémoniques (chaînes de caractères associées à une valeur d'entier) sont considérées comme faisant partie du type. Lorsqu'une énumération est reliée au terminal de sélection d'une structure Condition, les conditions sont nommées en fonction des mnémoniques de l'énumération plutôt que des valeurs numériques traditionnelles. La représentation numérique d'une énumération est toujours un octet, mot ou long, non signé.

Vous pouvez sélectionner des couleurs dans la *constante boîte de couleurs* afin de les utiliser avec la commande boîte de couleurs, une commande dont les valeurs sont en corrélation avec des couleurs spécifiques. Définissez la boîte de couleurs en cliquant dessus avec l'outil Pinceau ou l'outil Doigt et choisissez la couleur que vous voulez dans la palette de couleurs.

La constante *menu déroulant de symboles* sert à attribuer des symboles aux éléments d'une commande de type liste.

La *constante liste déroulante d'erreurs* est une liste déroulante prédéfinie. Vous pouvez cliquer sur la constante avec l'outil Doigt et sélectionnez le message d'erreur que vous voulez dans la boîte de dialogue qui s'affiche, comme le montre l'illustration suivante. Cette liste déroulante est utile avec les fonctions qui retournent des codes d'erreurs ou des clusters d'erreurs, telles que les fonctions E/S sur fichiers, car elle rend les diagrammes plus descriptifs. Par exemple, si vous essayez d'ouvrir un fichier inexistant en utilisant la fonction "Ouvrir un fichier", la fonction retourne alors un code d'erreur. Vous pouvez tester cette condition en comparant le code d'erreur de sortie à une liste déroulante d'erreurs définie à une valeur de Fichier introuvable (7).



Vous pouvez utiliser la *constante chemin* afin de créer une valeur de constante chemin sur le diagramme.

Les constantes chemin, chaîne de caractères et boîtes de couleurs sont redimensionnables, alors que les constantes booléennes, numériques, menu déroulant et énumération ne le sont pas. Une constante numérique adapte sa représentation à n'importe quelle valeur que vous entrez. Par exemple, si vous tapez 1.1 la constante devient DBL. Cette opération est contrôlée par le paramètre **Adapter à la source** du menu local **Représentation**.

La représentation "Adapter à la source" indique si une constante numérique détermine automatiquement sa représentation lorsque vous tapez une nouvelle valeur. Les constantes numériques du diagramme sont définies par défaut avec l'option "Adapter à la source". Par exemple, quand vous tapez 2 dans une constante numérique DBL, elle change automatiquement son type en I32. Si vous modifiez la représentation d'une constante par le biais du menu local, elle désactive de manière implicite l'option **Adapter à la source** afin que la constante reste du type spécifié. Si vous voulez restaurer l'ancien comportement, vous devez sélectionner **Adapter à la source** dans le menu local de la représentation.

La représentation par défaut d'une constante numérique est un nombre à virgule flottante double précision si vous entrez un nombre à virgule flottante, un entier long si vous entrez un entier, ou un nombre complexe double précision si vous entrez un nombre complexe. Par exemple, la représentation est un entier long si vous entrez '123' et un nombre à virgule flottante double précision si vous entrez '123.'. Vous pouvez modifier la représentation avec l'élément **Représentation** du menu local de la constante.

Constantes universelles

Les constantes universelles sont de deux types : constante numérique universelle et constante chaîne de caractères universelle.

- Constante numérique universelle : jeu de valeurs mathématiques et physiques précises et habituellement utilisées, telles que pi et la vitesse de la lumière.
- Constante chaîne de caractères universelle : jeu de chaînes de caractères habituellement non-affichables, telles que le saut de ligne et le retour chariot.

Pour plus d'informations sur ces constantes, consultez les références en ligne de LabVIEW ou de BridgeVIEW, en sélectionnant **Aide» Référence en ligne»Function and VI Reference**.

Nœuds

Les nœuds sont les éléments de l'exécution d'un diagramme. Les six types de nœuds sont les *fonctions*, *sous-VIs*, *structures*, *Code Interface Nodes* (CINs), *Formula Nodes* et *Attribute Nodes*. Les nœuds de fonctions et les nœuds de sous-VIs ont une apparence et un rôle identiques dans un diagramme, mais ils se différencient de manière substantielle. Les fonctions sont traitées dans la section *Fonctions* de ce chapitre. Les sous-VIs le sont dans le chapitre 3, *Utilisation des sous-VIs*.

Les structures, qui complètent le modèle de programmation à flux de données du G afin de contrôler l'ordre de l'exécution, sont brièvement traitées dans la section *Structures* de ce chapitre. Pour obtenir des informations plus détaillées, consultez le chapitre 19, *Structures*.

Les CIN sont des interfaces entre le diagramme et des codes écrits dans des langages de programmation conventionnels, tels que le C ou le Pascal. Pour plus d'informations, consultez le *LabVIEW Code Interface Reference Manual* disponible au format PDF sur les disquettes ou CD du logiciel.

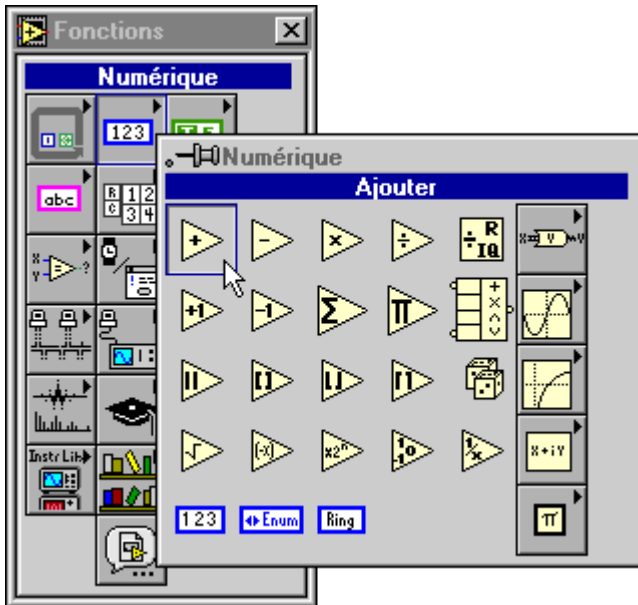
Les boîtes de calcul, qui complètent les fonctions du G en utilisant des formules sur le diagramme, sont traitées dans la chapitre 20, *Boîtes de calcul*.

Les attribute nodes, qui modifient les attributs de commande par programme, sont traités dans le chapitre 22, *Attribute Nodes*.

Fonctions

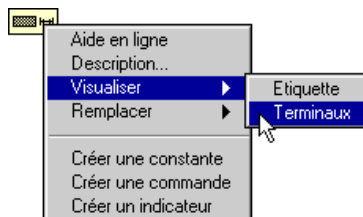
Les fonctions sont des nœuds élémentaires intégrés au G. Elles exécutent des opérations élémentaires telles que l'addition des nombres, les E/S sur fichiers et le formatage des chaînes de caractères. Les fonctions n'ont pas de faces-avant ni de diagrammes. Une fois compilées, elles génèrent un code machine en ligne. Pour obtenir des descriptions détaillées sur chaque fonction, consultez **Référence en ligne»Function and VI Reference**.

Vous pouvez sélectionner les fonctions dans la palette **Fonctions**, comme le montre l'illustration suivante.



Lorsque vous sélectionnez une fonction, son icône s'affiche sur le diagramme. Pour afficher son étiquette, ouvrez un menu local de l'icône et sélectionnez **Visualiser**»**Etiquette**. Vous pouvez modifier l'étiquette en mettant en surbrillance le texte avec l'outil Texte et en le retapant. Vous pouvez utiliser l'étiquette de la fonction pour annoter son rôle dans le diagramme.

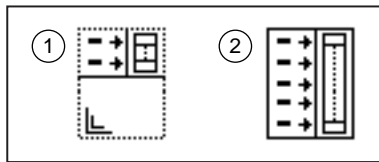
Vous pouvez utiliser la fenêtre d'aide pour comprendre comment câbler la fonction ou sélectionner **Visualiser**»**Terminaux** dans le menu local de l'icône pour voir l'emplacement précis des terminaux, comme le montre l'illustration suivante.



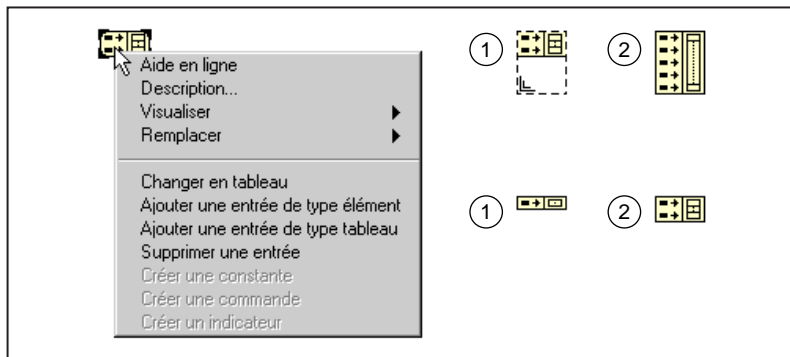
Lorsque les terminaux sont affichés, sélectionnez à nouveau **Visualiser» Terminaux** pour afficher l'icône de la fonction.

Certaines fonctions ont un nombre de terminaux variable. Par exemple, si vous concevez un tableau de trois éléments, la fonction "Construire un tableau" nécessite trois terminaux d'entrée, mais si vous en concevez un autre avec dix éléments, il vous faudra dix terminaux.

Vous pouvez modifier le nombre de terminaux des fonctions *extensibles* en redimensionnant l'icône avec le curseur de redimensionnement de manière identique aux autres objets, comme le montre l'illustration suivante. Vous pouvez agrandir ou rétrécir les fonctions, mais vous ne pouvez pas rétrécir une fonction si le rétrécissement doit provoquer la disparition d'un terminal câblé.



Vous pouvez également modifier le nombre de terminaux avec les commandes **Ajouter une entrée** et **Supprimer une entrée** du menu local d'un terminal, comme le montre l'illustration suivante. La commande **Supprimer une entrée** supprime le terminal sur lequel vous avez ouvert un menu local et déconnecte le fil, si le terminal est câblé. La commande **Ajouter une entrée** ajoute un terminal immédiatement après le terminal sur lequel vous avez ouvert un menu local. Les noms complets de ces commandes varient selon la fonction.



Structures

Lorsque vous programmez, il arrive parfois qu'il soit nécessaire de répéter des sections de code un nombre de fois donné ou tant qu'une certaine condition reste vraie. Dans d'autres situations, il est nécessaire d'exécuter différentes sections de code lorsque différentes conditions existent, ou d'exécuter un code dans un ordre spécifique. Le G possède quatre nœuds spéciaux, appelés *structures*. Ils vous aident à contrôler l'exécution du code dans des situations où il n'est pas possible de le faire dans la structure du flux de données du G.

Chaque structure possède une bordure distincte et redimensionnable que vous pouvez utiliser afin d'encadrer le code qui s'exécute en fonction des règles spéciales de la structure. C'est pour cette raison que le diagramme qui se situe à l'intérieur de la bordure de la structure est appelé un sous-diagramme. Vous pouvez imbriquer plusieurs sous-diagrammes les uns dans les autres.

Le G utilise les structures suivantes pour l'exécution du code :

- Boucle For : répète l'exécution un certain nombre de fois.
- Boucle While : répète l'exécution de son sous-diagramme tant qu'une condition reste vraie (TRUE).
- Structure Condition : contient de nombreux sous-diagrammes parmi lesquels un seul s'exécute en fonction de la valeur transmise à son terminal sélecteur.
- Structure Séquence : exécute le code dans l'ordre numérique de ses sous-diagrammes.

Etant donné que les structures sont des nœuds, elles ont des terminaux qui les connectent à d'autres nœuds. Par exemple, les terminaux qui font passer des données à l'intérieur et à l'extérieur des structures sont appelés *tunnels*. Les tunnels sont des points de connexion délocalisables pour les fils de liaison à partir de l'intérieur et de l'extérieur des structures. Pour plus d'informations, consultez le chapitre 19, *Structures*.

Remplacement et insertion des objets du diagramme

Supposons que vous souhaitiez utiliser une fonction **Incrément** dans un diagramme alors que la fonction **Décrément** est celle qui est utilisée. Vous pouvez supprimer le nœud de la fonction **Incrément** et ensuite sélectionner le nœud **Décrément** de la palette **Fonctions** et la câbler à nouveau. Vous pouvez également utiliser l'élément **Remplacer** du menu local du nœud **Incrément**. Le fait de sélectionner **Remplacer** donne accès à la palette **Fonctions** à partir de laquelle vous pouvez choisir la fonction **Décrément**. L'avantage de cette méthode est que le nouveau nœud remplace l'ancien et ne modifie pas le câblage. Vous pouvez remplacer une fonction par une autre fonction, bien que si le nombre de terminaux ou de types de données de chaque nœud de fonction est différent, vous risquez d'avoir des fils brisés.

Vous pouvez également utiliser la fonction **Remplacer** pour remplacer une constante par une autre constante ou une structure par une autre structure similaire, telle qu'une boucle **While** par une boucle **For**.



Remarque

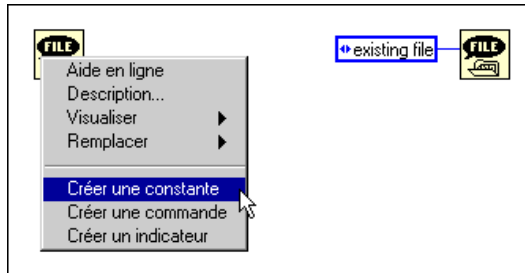
*Lorsque vous utilisez **Remplacer** sur un sous-VI, si vous sélectionnez un VI dont le nom est identique à un autre qui se trouve déjà en mémoire, le nœud remplacé fera référence à celui qui se trouve déjà en mémoire et non pas au VI sélectionné.*

Les menus locaux des fils ont un élément **Insérer**. Si vous l'ouvrez, vous accédez à la palette **Fonctions**. A partir de cette dernière, vous pouvez choisir n'importe quel VI ou fonction sur la palette. L'éditeur raccorde alors le nœud que vous choisissez au fil sur lequel vous avez ouvert un menu local. Toutefois, vous devez bien vérifier le câblage si le nœud possède plus d'un terminal d'entrée ou de sortie car les fils peuvent ne pas être connectés au terminal voulu.

Ajout automatique de constantes, commandes et indicateurs

Au lieu de créer une constante, une commande ou un indicateur en le sélectionnant dans un menu et en le câblant manuellement à un terminal, vous pouvez ouvrir un menu local sur un terminal et choisir **Create Constant**, **Créer une commande** ou **Créer un indicateur** afin de créer automatiquement un objet avec un type de données approprié. La nouvelle constante, commande ou le nouvel indicateur créé est automatiquement câblé. Si le terminal correspond à une définition de type, vous créez alors une constante, une commande ou un indicateur de même définition de type. Si la définition de type est modifiée par la suite, l'objet est alors mis à jour.

Par exemple, si vous avez besoin d'une constante pour le **mode de sélection d'entrée** de la fonction "Boîte de dialogue de fichier", vous pouvez ouvrir un menu local sur l'entrée et sélectionner **Create Constant**. L'éditeur crée à votre place un type énuméré qui contient toutes les valeurs autorisées pour le **mode de sélection**. En supposant que le mode d'entrée n'est pas déjà câblé, la nouvelle constante sera automatiquement câblée.



Les sorties de fonctions ou de VIs, les fils, les constantes et les terminaux de commandes ou indicateurs de face-avant vous permettent également de **Create Constant**, **Créer une commande** ou **Créer un indicateur**.

Câblage du diagramme

Ce chapitre présente la manière de connecter des terminaux sur le diagramme en les reliant les uns aux autres.

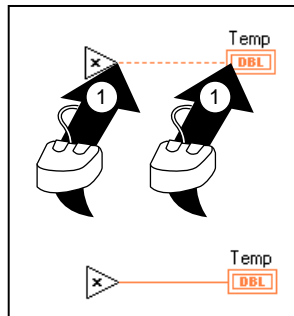
Techniques de câblage

Vous devez utiliser l'outil Bobine pour connecter les terminaux entre eux. Le pointeur, ou l'amorce, de l'outil est le bout de la portion du fil déroulé, comme le montre l'illustration suivante.

amorce du curseur de câblage



Le symbole illustré à gauche représente la souris. Dans les illustrations de câblage de ce chapitre, la flèche à l'extrémité du symbole de la souris montre l'endroit où il faut cliquer, et le nombre marqué sur le bouton de la souris indique le nombre de fois qu'il faut cliquer.

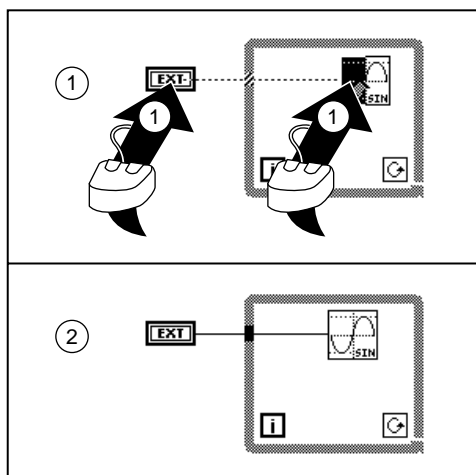


Pour relier un terminal à un autre, cliquez avec l'outil Bobine sur le premier terminal, déplacez l'outil vers le deuxième terminal et cliquez ensuite sur le deuxième terminal, comme le montre la précédente illustration. Ne maintenez pas le bouton de la souris enfoncé lors du déplacement de la souris. Le fait de cliquer en premier sur un des deux terminaux n'a pas d'importance. La zone du terminal clignote lorsque l'amorce de l'outil Bobine se positionne correctement sur le terminal. En cliquant sur ce terminal, vous lui connectez un fil de liaison. Lors de la première

connexion, un fil de liaison se dessine entre le terminal et l'outil au fur et à mesure que vous déplacez le curseur sur le diagramme, comme si le fil se déroulait d'une bobine. Vous n'avez pas besoin de maintenir le bouton de la souris enfoncé.

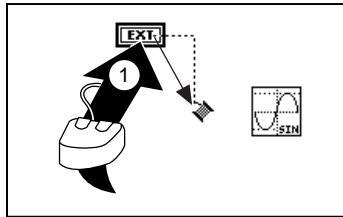
Pour effectuer une connexion à partir d'un fil de liaison existant, exécutez l'opération décrite plus haut, en la commençant ou la terminant sur le fil de liaison existant. Le fil de liaison clignote lorsque l'outil Bobine est correctement positionné pour attacher un nouveau fil au fil existant.

Vous pouvez directement relier un terminal en dehors de la structure à un terminal dans la structure en utilisant l'opération de câblage de base. Un *tunnel* est automatiquement créé lorsque le fil de liaison traverse la limite de la structure, comme le montre l'illustration suivante.

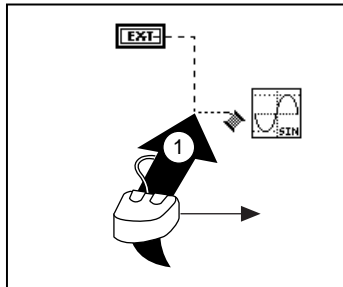


Les fils de liaison se déroulent verticalement ou horizontalement à partir des terminaux, en fonction de la première direction dans laquelle vous déplacez l'outil Bobine. Les fils de liaison se déplacent verticalement si vous déplacez l'outil vers le haut ou vers le bas et horizontalement si vous déplacez l'outil vers la gauche ou la droite. La connexion est centrée sur les terminaux, quelle que soit la position exacte de l'amorce au moment où

vous cliquez sur le bouton de la souris, comme le montre l'illustration suivante.



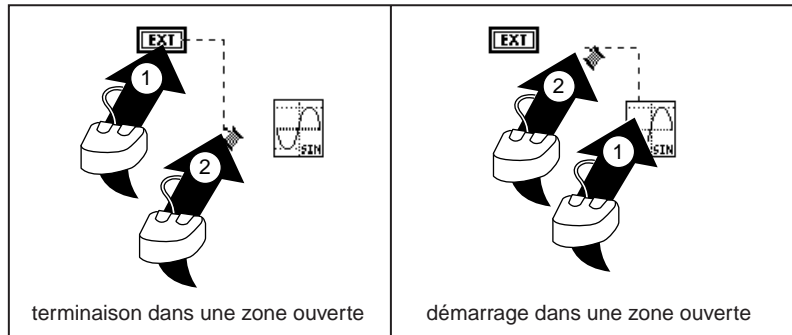
Vous pouvez couder vos fils de liaison à 90° en une seule fois sans devoir cliquer. Vous pouvez cliquer sur une zone libre pour fixer le fil de liaison et lui faire changer de direction, comme le montre l'illustration suivante.



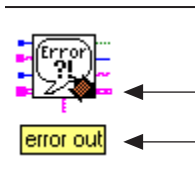
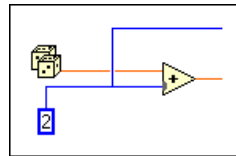
Remarque

Vous pouvez également changer de direction en appuyant sur la barre d'espace. Vous pouvez décrocher la dernière fixation en appuyant sur <Ctrl-clic> (Windows), <option-clic> (Macintosh), <bouton du milieu-clic> (UNIX). Si la dernière fixation est un terminal ou un fil de liaison sur lequel vous avez d'abord cliqué, le décrochage supprime tout le fil de liaison.

Vous pouvez double-cliquer avec l’outil Bobine pour commencer ou terminer un fil de liaison dans une zone libre, comme le montre l’illustration suivante.



Lorsque des fils de liaison se croisent, un petit intervalle s’affiche sur le premier fil dessiné, comme s’il se trouvait sous le deuxième fil, comme dans l’illustration ci-dessous.

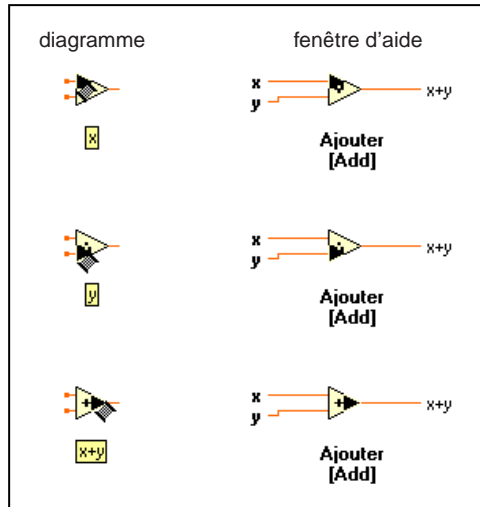


Lorsque vous câblez une fonction ou un sous-VI compliqué, vous devez faire attention aux amorces de connexion et vous reporter aux info-bulles qui s’affichent quand l’outil Bobine s’approche de l’icône du VI.

Les amorces de connexion, les fils de liaison tronqués illustrés à gauche de l’icône du VI, indiquent le type de données de chaque terminal par leur style, épaisseur, et couleur (Pour plus de détails, consultez *Carte de référence rap de programmation en G*). Les points à l’extrémité des amorces de connexion indiquent les entrées. Les sorties n’ont pas de points. La direction dans laquelle les amorces de connexion sont dessinées indique la direction du câblage suggéré afin de produire des diagrammes propres. Lorsque vous câblez un terminal, l’amorce de connexion disparaît pour ce terminal.

L’info-bulle, la boîte illustrée à gauche et qui contient du texte, indique le nom du terminal auquel vous vous reliez lorsque vous appuyez sur le bouton de la souris.

Vous pouvez également profiter de la fonction de la fenêtre **Aide** qui met en surbrillance chaque terminal du cadre connecteur. Grâce à cette fonction, vous pouvez voir exactement où les fils de liaison doivent être connectés. Les trois connexions de la fonction Ajouter sont représentées dans l'illustration suivante.

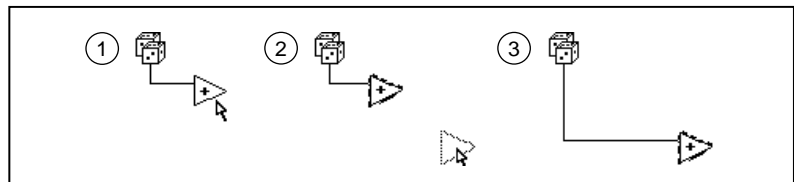


 **Remarque**

La fonction de fenêtre d'aide illustrée ci-dessus n'est pas supportée par les fonctions qui peuvent être agrandies. La fonction "Assembler un tableau" en est un exemple.

Adaptation des liaisons

Vous pouvez déplacer des objets câblés de manière individuelle ou en groupe en faisant glisser les objets sélectionnés vers leur nouvel emplacement avec l'outil Flèche. Les fils de liaison reliés aux objets sélectionnés s'adaptent automatiquement. L'illustration suivante vous montre la fonction d'adaptation des liaisons.



Si vous dupliquez les objets sélectionnés ou les déplacez d'un diagramme à un autre – par exemple, du diagramme vers un sous-diagramme de

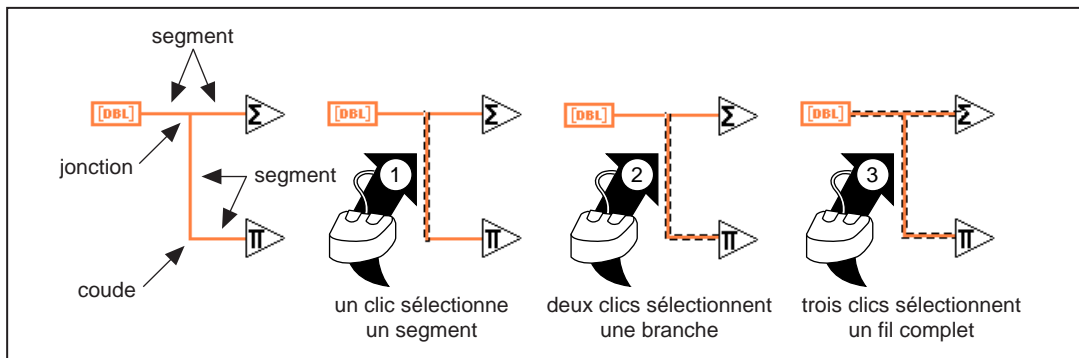
structure – les fils de liaison qui y sont reliés sont coupés, à moins que vous ne les sélectionniez aussi.

 **Remarque**

L'adaptation des liaisons provoque parfois des amorces de connexion ou des segments déconnectés. Ce problème est traité dans la sous-section Segments déconnectés de la section Résolution des problèmes de câblage de ce chapitre. Vous devez supprimer les Amorces de connexions ou les segments déconnectés pour que le VI s'exécute. La manière la plus facile de le faire consiste à sélectionner la commande Edition»Supprimer les fils incorrects. Cette opération supprime également les boucles superflues dans les fils de liaison.

Sélection, déplacement et suppression des fils de liaison

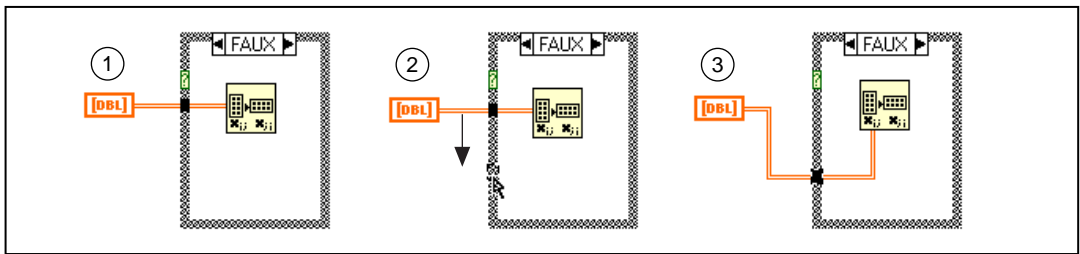
Une portion de fil est un simple morceau de fil horizontal ou vertical. Le point auquel trois ou quatre portions de fils se rejoignent est une *jonction*. Un *coude* dans un fil est l'endroit où deux portions se rejoignent. Une *branche* de fil contient toutes les portions de fil d'une jonction à une autre, d'un terminal à une jonction, d'un terminal à un autre s'il n'y a aucune jonction entre eux. Un clic de souris avec l'outil Flèche sur un fil de liaison sélectionne une portion. Deux clics sélectionnent une branche. Trois clics sélectionnent un fil tout entier. Appuyez sur la touche <Suppr> ou <Retour arrière> pour supprimer la portion de fil sélectionnée. Cette opération vous est montrée dans l'illustration suivante.



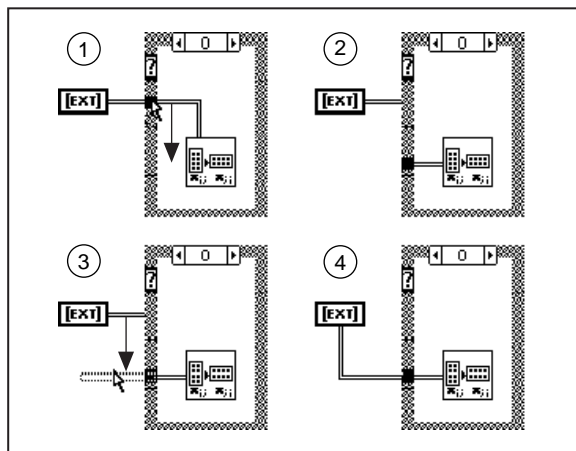
Le repositionnement d'un objet câblé peut créer une portion supplémentaire ou une portion de fil bizarre. Pour repositionner une portion de fil, faites-la glisser vers son nouvel emplacement avec l'outil Flèche. Vous pouvez repositionner une ou plusieurs portions en les sélectionnant et en les faisant glisser. Utilisez la touche <Maj> pour restreindre le déplacement du fil à un déplacement horizontal ou vertical uniquement. La direction dans laquelle vous démarrez le déplacement détermine si le fil de

liaison est limité à une translation horizontale ou verticale. Vous pouvez également déplacer les portions sélectionnées pixel par pixel en appuyant sur les touches fléchées du clavier. Maintenez la touche <Ma j> enfoncée quand vous utilisez les touches fléchées pour déplacer les portions sélectionnées de plusieurs pixels à la fois. Les portions adjacentes et non sélectionnées s'adaptent à la modification. Vous pouvez simultanément sélectionner et faire glisser plusieurs portions de fil, même si elles sont discontinues.

Lorsque vous déplacez un tunnel, une connexion du fil est normalement conservée entre le tunnel et le nœud câblé, comme le montre l'illustration suivante.



Toutefois, le déplacement d'un tunnel provoque parfois la création d'une portion de fil supplémentaire qui se situe sous la bordure de la structure. Vous ne pouvez pas sélectionner et faire glisser cette portion car elle est masquée, mais elle disparaît si vous faites glisser la portion qui lui est reliée, comme le montre l'illustration suivante. Si vous n'êtes pas sûr du fil de liaison précis qui est relié au tunnel, cliquez trois fois sur le fil de liaison.



Pour sélectionner à la fois les parties d'un fil de liaison qui se trouvent à l'intérieur et à l'extérieur d'une structure en boucle, sélectionnez la partie du fil d'un côté de la structure et maintenez la touche <Ma j> enfoncée lorsque vous sélectionnez la partie du fil de l'autre côté. Vous pouvez ajouter un objet à un groupe d'objets précédemment sélectionnés en maintenant la touche <Ma j> enfoncée au moment de la sélection du nouvel objet. De même, vous pouvez faire glisser un rectangle de sélection autour des deux parties du fil de liaison. La structure ne sera pas sélectionnée tant que vous ne l'entourez pas complètement avec le rectangle de sélection. Les autres nœuds sont sélectionnés s'ils touchent le rectangle.

Câblage hors écran

Vous aurez peut-être à relier un objet à l'écran à un autre qui se trouve en dehors de l'écran. Ceci se produit lorsque, par exemple, les objets sont trop éloignés les uns des autres pour apparaître ensemble à l'écran. Pour cela, cliquez d'abord avec l'outil Bobine sur un terminal de l'objet à l'écran. Ensuite, vous pouvez faire automatiquement défiler le diagramme lors du câblage en faisant glisser l'outil Bobine légèrement au-delà du bord du diagramme. Lorsque le deuxième objet apparaît, cliquez sur un de ses terminaux pour effectuer le câblage.

Résolution des problèmes de câblage

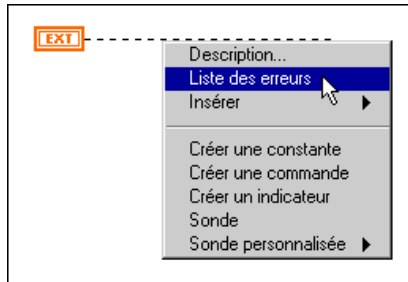
Fils incorrects

- - -



Lorsque vous réalisez une erreur de câblage, un *fil de liaison brisé* signalé par une ligne pointillée peut s'afficher. Parfois, vous avez une connexion de câblage erronée, mais elle reste invisible car la portion du fil de liaison brisé est très petite ou masquée par un objet. Si le bouton **Exécution** affiche une flèche brisée (illustré à gauche) mais que vous ne voyez aucun problème dans le diagramme, sélectionnez **Edition»Supprimer les fils incorrects**. Si le bouton **Exécution** affiche une flèche en bon état (illustré à gauche), c'est que vous avez corrigé le problème. Dans le cas contraire, cliquez sur le bouton **Flèche brisée** pour obtenir une liste des erreurs.

Si vous ne savez pas pourquoi un fil de liaison donné s'est brisé, ouvrez un menu local et choisissez **Liste des erreurs**, comme le montre l'illustration suivante. La boîte de dialogue **Liste des erreurs** s'affiche et répertorie toutes les erreurs. Cliquez sur une des erreurs et le fil de liaison en faute sera sélectionné, ce qui vous permettra alors de le supprimer ou de le réparer.



Pour plus d'informations, consultez les sections *Réparation des VIs invalides* et *Mise au point des VIs exécutables* du chapitre 4, *Exécution et mise au point des VIs et sous-VIs*.

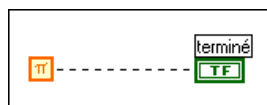
La liste suivante répertorie certaines erreurs possibles de câblage.

- Fil de liaison : conflit de type
- Fil de liaison : conflit de dimension
- Fil de liaison : conflit d'élément
- Fil de liaison : conflit de refnum
- Fil de liaison : conflit de classe
- Fil de liaison : conflit d'énumération
- Fil de liaison : conflit d'unités
- Fil de liaison : a plusieurs sources
- Fil de liaison : n'a pas de source
- Fil de liaison : a des segments déconnectés
- Fil de liaison : fait partie d'un cycle

Pour plus d'informations sur les structures de câblage, consultez la section *Problèmes de câblage de structure* du chapitre 19, *Structures*.

Conflits de type pour les fils de liaison

Une incohérence de type survient lorsque vous reliez deux objets de types de données différents, tels qu'un numérique et un booléen comme le montre l'illustration suivante.



Les erreurs de conflit de dimension et de conflit d'élément surviennent dans des situations identiques – respectivement lorsque vous reliez deux tableaux dont les éléments correspondent mais pas les dimensions et lorsque vous reliez deux clusters dont les éléments ont des différences de type.

Le conflit de refnum résulte du câblage de deux types de refnums différents ou de deux refnums de fichiers d'enregistrement de données ayant des types d'enregistrements différents.

L'erreur de conflit de classe se produit lorsque vous reliez deux refnums du même type mais qui ont des classes différentes.

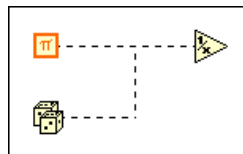
L'erreur de conflit d'énumération survient lorsque vous reliez deux énumérations incompatibles. Deux énumérations sont compatibles si elles ont exactement le même jeu de chaînes de caractères ou si l'une d'entre elles a un jeu de chaînes de caractères identique aux premières chaînes de caractères de l'autre.

Le conflit d'unités survient lorsque vous reliez deux objets qui n'ont pas d'unités comparables.

Toute autre incohérence résulte d'une erreur de conflit de type. Ce problème survient généralement lorsque vous connectez un fil de liaison au mauvais terminal d'une fonction ou d'un sous-VI. Sélectionnez et supprimez le fil de liaison et reliez-le à nouveau au bon terminal. Vous pouvez utiliser la fenêtre d'aide pour éviter ce type d'erreurs. Parfois, vous devrez modifier le type d'un des terminaux.

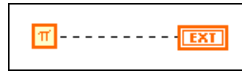
Plusieurs sources de fil de liaison

Vous pouvez relier une simple source de données à plusieurs destinations, mais vous ne pouvez pas relier plusieurs sources de données à une seule destination. Dans l'exemple suivant, vous devez déconnecter une source.



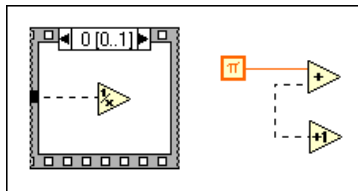
Pendant l'élaboration de la face-avant, vous avez peut-être déposé une commande alors que vous vouliez déposer un indicateur. Si vous essayez de relier une valeur de sortie au terminal d'une commande de face-avant, vous obtiendrez une erreur de sources multiples. Pour corriger l'erreur,

ouvrez le menu local du terminal et sélectionnez la commande **Changer en indicateur**.



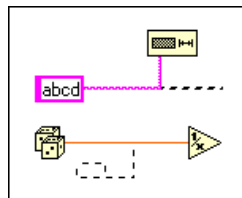
Absence de source de liaison

Deux exemples de fils de liaison n'ayant pas de source vous sont proposés dans l'illustration suivante. Dans l'un, un tunnel fournit des données à la fonction Inverser, mais rien ne fournit de données au tunnel. Dans l'autre, deux entrées de fonctions sont reliées, mais il n'y a aucune source de données pour elles. La solution à ces problèmes consiste à respectivement relier une source de données au tunnel ainsi qu'aux fonctions Ajouter et Incrément. Vous obtiendrez également cette erreur si vous reliez deux terminaux d'indicateurs de face-avant alors qu'un des deux doit être une commande. Dans ce cas, ouvrez le menu local sur un terminal et sélectionnez l'élément **Changer en commande**.



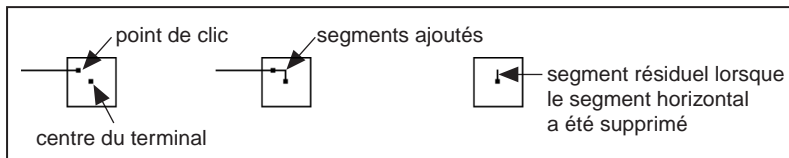
Segments déconnectés

Les *segments déconnectés*, comme le montre l'illustration suivante, sont des branches de fil de liaison qui ne sont pas connectées à un terminal. Ceci peut être dû à l'adaptation des liaisons, à un retraçage durant l'opération de câblage ou à la suppression des objets câblés. Sélectionnez **Edition» Supprimer les fils incorrects** pour arranger les segments déconnectés.



Lorsque vous vous connectez à un terminal, vous cliquez rarement sur le terminal lorsque le curseur se situe au centre. Par conséquent, lorsque vous n'êtes pas sur le centre, une portion de fil de liaison relie automatiquement

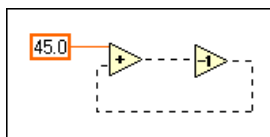
le point sur lequel vous avez cliqué au centre du terminal. Si vous supprimez par la suite le fil de liaison relié au terminal, un segment déconnecté restera peut-être, ce qui provoquera un bouton Flèche brisée. L'illustration suivante vous montre ce problème.



Pour éviter cette situation, cliquez trois fois sur le fil de liaison pour vous assurer que toutes les portions sont sélectionnées avant de le supprimer. Vous pouvez également utiliser **Edition»Supprimer les fils incorrects** pour supprimer les segments déconnectés.

Cycles de fils de liaison

Les fils de liaison ne doivent pas former de cycle. C'est-à-dire, les fils de liaison ne doivent pas former de boucle fermée d'icônes ou de structures comme le montre l'illustration suivante. Le système d'exécution ne peut exécuter des cycles car chaque nœud attend que l'autre lui fournisse des données avant de s'exécuter.



La section *Registres à décalage* du chapitre 19, *Structures*, décrit la bonne manière de renvoyer des données dans un calcul répétitif.

Situations de câblage à éviter

Les sections suivantes décrivent des situations qui ne produisent pas de fils incorrects mais qui rendent la lecture du diagramme difficile ou qui rendent son aspect trompeur quant à ses actions.

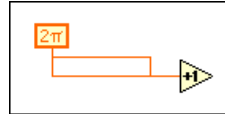


Remarque

Souvenez-vous que lorsque vous n'êtes pas sûr de ce qui est relié à un fil de liaison, vous pouvez double-cliquer sur le câble pour sélectionner la branche ou cliquer trois fois pour sélectionner l'ensemble du fil de liaison.

Fils de liaison en boucle

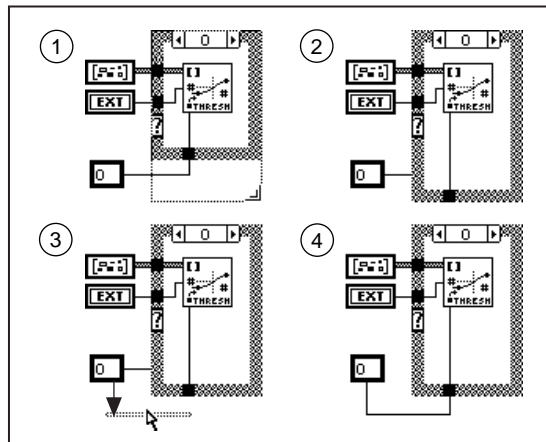
Une boucle de fils de liaison n'est pas une erreur mais une mauvaise conception car elle alourdit le diagramme sans raison. Cliquez deux fois sur une des branches pour la sélectionner et supprimez-la. L'illustration suivante vous montre un fil de liaison en boucle.



La commande **Edition»Supprimer les fils incorrects** supprimera une des branches de ces boucles afin de nettoyer les fils de liaison.

Portions de fil masquées

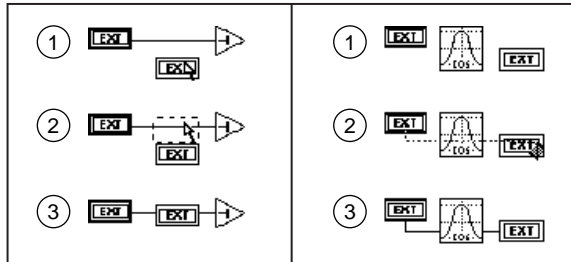
Essayez d'éviter le câblage sous la bordure d'une structure ou entre des objets superposés car certaines portions du fil de liaison créé peuvent être masquées. L'illustration suivante vous montre un exemple de portions de fil de liaison masquées.



Vous pouvez créer des portions de fil de liaison masquées par mégarde, lorsque par exemple vous déplacez un tunnel ou agrandissez une structure, comme dans les cas 1 et 2 de la précédente illustration. Les cas 3 et 4 de cet exemple montrent une manière de réduire l'ampleur du désordre dans ce diagramme. Vous pouvez faire glisser la portion du fil de liaison reliée à la constante de manière à ce qu'elle réapparaisse. Appuyez sur la touche **<Maj>** pour forcer le glissement du fil de liaison et diminuer les risques de création de segments déconnectés.

Câblage sous des objets

Les fils de liaison ne relient que les objets sur lesquels vous cliquez. Le fait de faire glisser un terminal ou une icône sur un fil de liaison donne l'impression qu'il est relié alors qu'il ne l'est pas, comme le montre l'illustration suivante dans la partie gauche (1, 2, 3 à gauche).

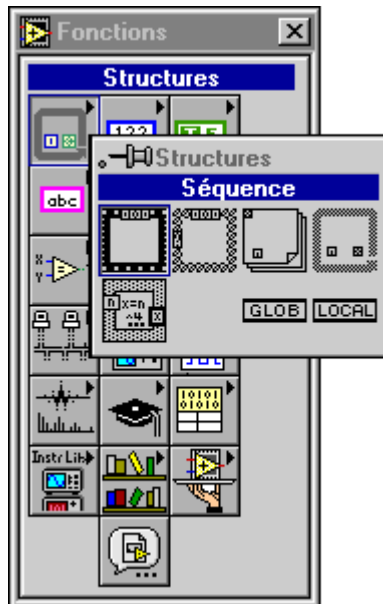


Le fait de faire glisser un fil de liaison par-dessus une icône ou un terminal donne la même impression alors qu'il n'est que derrière l'icône, comme le montre l'illustration dans la partie droite (1, 2, 3 à droite). Evitez ces situations car elles sont à l'origine de confusions visuelles.

Pour plus d'informations sur ce problème, consultez la section *Comprendre les mises en garde* du chapitre 4, *Exécution et mise au point des VIs et sous-VIs*.

Structures

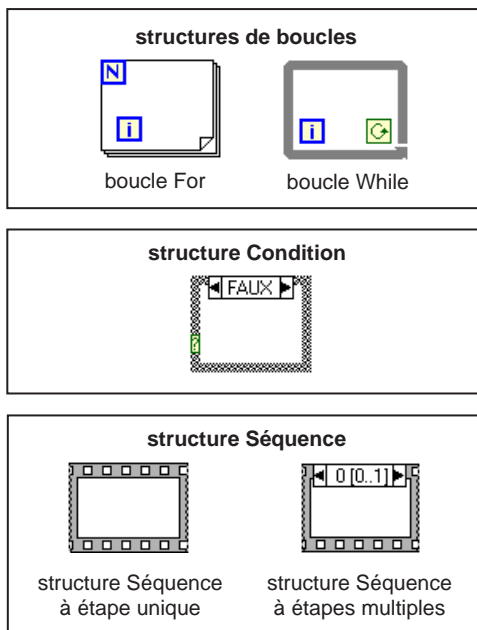
Ce chapitre présente l'utilisation des boucles For et While et des structures Condition et Séquence. Ces structures se trouvent dans la palette **Fonctions»Structures**, comme le montre l'illustration suivante.



Consultez la bibliothèque `exemples\general\structs.llb` pour des exemples d'utilisation de ces structures.

Les structures sont des nœuds qui complètent le flux de l'exécution d'un diagramme, comme les structures de commande d'un langage de programmation conventionnel. L'icône de chaque structure du G est une

boîte redimensionnable ayant une bordure distincte, comme le montre l'illustration suivante.



Les structures se comportent comme les autres nœuds dans le sens où elles s'exécutent automatiquement dès que des données d'entrée sont disponibles et ne fournissent des données à leurs fils de liaison de sortie qu'à la fin de l'exécution. Toutefois, chaque structure exécute son sous-diagramme en fonction des règles décrites dans les sections suivantes.

Un sous-diagramme est une série de nœuds, de fils de liaison et de terminaux qui résident à l'intérieur de la bordure de la structure. La boucle For et la boucle While possèdent chacune un sous-diagramme. Toutefois, les structures Condition et Séquence peuvent posséder plusieurs sous-diagrammes empilés comme un jeu de cartes où une seule est visible à la fois. Vous pouvez construire les sous-diagrammes de la même manière qu'un diagramme de haut niveau ; les sous-diagrammes peuvent contenir des terminaux, des nœuds (y compris d'autres structures) et des fils de liaison de diagramme.

Vous devez créer des terminaux pour l'entrée et la sortie automatiques des données de la structure pour que des fils de liaison traversent la limite de la structure. Ces terminaux de limite sont appelés des *tunnels*. Les tunnels ont toujours un côté à l'intérieur de la structure et un autre à l'extérieur. Un

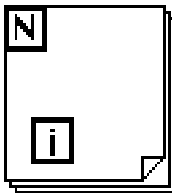
tunnel se situe toujours sur la bordure de la structure mais vous pouvez le déplacer à n'importe quel endroit le long de cette bordure en le faisant glisser avec l'outil Flèche.

Les structures possèdent également d'autres terminaux particuliers à chaque type de structure. Ces terminaux sont décrits avec les structures correspondantes dans les sections suivantes.

Structures de boucle For et de boucle While

Vous pouvez utiliser la boucle For et la boucle While afin de contrôler des opérations répétitives, jusqu'à un nombre spécifié d'itérations (boucle For) ou tant qu'une condition spécifiée reste vraie (boucle While).

Boucle For



Terminal d'itération



Terminal de comptage

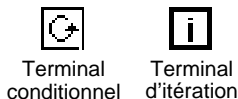
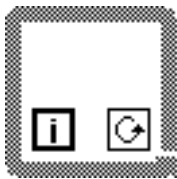
Une boucle For exécute son sous-diagramme un certain *nombre* de fois, où le nombre équivaut à la valeur contenue dans le *terminal de comptage*. Vous pouvez explicitement définir le nombre en reliant une valeur de l'extérieur de la boucle au côté gauche ou supérieur du terminal de comptage ou avec l'*auto-indexation* (pour plus d'informations, consultez la section *Auto-indexation* de ce chapitre). Les autres côtés du terminal de comptage sont transmis à l'intérieur de la boucle afin que vous puissiez accéder au nombre de manière interne.

Le *terminal d'itération* contient le nombre d'itérations achevées en cours d'exécution ; 0 pendant la première itération, 1 pendant la deuxième, et ainsi de suite jusqu'à $N - 1$. *Le nombre et les terminaux d'itération sont des entiers signés longs allant de 0 à $2^{31} - 1$* . Si vous reliez un nombre à virgule flottante au terminal de comptage, le G l'arrondit si nécessaire et le contraint dans la gamme. Si vous reliez 0 au terminal de comptage, la boucle ne s'exécute pas.

La boucle For est équivalente au pseudo-code suivant :

```
for i = 0 to N - 1
    Execute subdiagram
```

Boucle While



Une boucle While exécute son sous-diagramme jusqu'à ce qu'une valeur booléenne que vous reliez au *terminal conditionnel* soit fausse (FALSE). Le G vérifie la valeur du terminal conditionnel à la fin de chaque itération et, si la valeur est vraie (TRUE), une autre itération se produit afin que la boucle s'exécute toujours au moins une fois. La valeur par défaut du terminal conditionnel est FALSE. Par conséquent s'il n'est pas câblé, la boucle ne s'effectue qu'une fois.

Le terminal d'itération contient le nombre d'itérations effectuées, comme dans la boucle For.

La boucle While est équivalente au pseudo-code suivant :

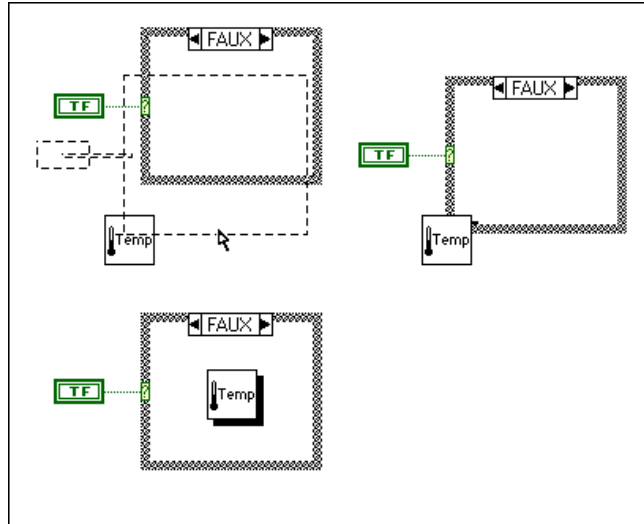
```
Do
    Execute subdiagram (which sets condition)
While condition is TRUE
```

Vous pouvez ajouter des terminaux appelés *registres à décalage* aux deux structures de boucle. Vous pouvez utiliser les registres à décalage pour transmettre des données de l'itération en cours à la suivante. Pour plus d'informations, consultez la section *Registres à décalage* de ce chapitre.

Placement d'objets à l'intérieur des structures

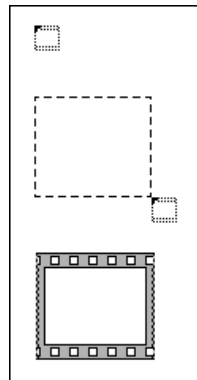
Vous pouvez placer un objet à l'intérieur d'une structure en le faisant glisser ou en construisant la structure autour de l'objet.

Vous ne pouvez pas placer un objet à l'intérieur d'une structure en faisant glisser la structure sur l'objet. Si vous déplacez une structure et qu'elle se superpose à un autre objet, l'éditeur affiche l'objet qu'il considère être en premier plan. Si vous mettez la structure entièrement sur un autre objet, et que l'éditeur en fait l'objet de premier plan, il affiche une ombre épaisse pour vous avertir que l'objet se trouve en dessous et non pas à l'intérieur de la structure. Sinon, la structure est complètement masquée par l'objet (pour plus d'informations sur les objets que votre application place en premier plan, consultez la section *Positionnement d'objets* du chapitre 2, *Edition des VIs*). L'illustration suivante vous montre les deux situations.



Placement et dimensionnement de structures sur le diagramme

Lorsque vous sélectionnez une structure à partir de **Fonctions»Structures** et que vous vous préparez à la mettre sur votre diagramme, vous pouvez voir une petite icône de la structure à la place de votre curseur. Cette icône est prête à être placée sur votre diagramme. Si vous cliquez et relâchez sans redimensionner, vous restez en mode de redimensionnement jusqu'à ce que vous cliquez à nouveau. Vous pouvez cliquer sur le diagramme et maintenir le bouton de la souris enfoncé, puis faire glisser la structure à la dimension voulue. L'illustration suivante vous en montre un exemple. Vous pouvez également dimensionner la structure *après* l'avoir déposée sur le diagramme en cliquant sur un des coins avec le curseur de redimensionnement, puis en le faisant glisser.



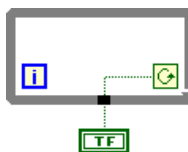
Placement de terminaux à l'intérieur des boucles

Les entrées transmettent les données dans la boucle au début de son exécution. Les sorties ne transmettent les données hors de la boucle qu'une fois toutes les itérations achevées.

Vous devez placer un terminal à l'intérieur d'une boucle lorsque vous voulez que la boucle vérifie la valeur du terminal à chaque itération. Par exemple, lorsque vous placez le terminal d'une commande booléenne de face-avant à l'intérieur d'une boucle While et que vous reliez le terminal au terminal conditionnel de la boucle, la boucle vérifie la valeur du terminal à *chaque* itération afin de déterminer si elle doit s'effectuer à nouveau. Vous pouvez arrêter cette boucle While, ci-après illustrée, en modifiant la valeur de la commande de face-avant en fausse (FALSE).



Si vous placez le terminal de la commande booléenne à l'extérieur de la boucle While, comme dans l'illustration suivante, vous provoquez une boucle infinie si la commande booléenne est vraie (TRUE) lorsque la boucle commence.

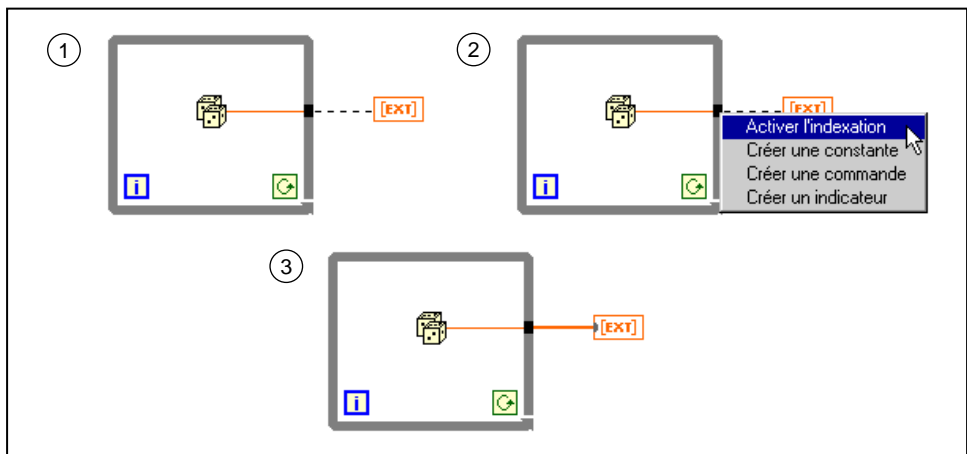


Si la commande est vraie (TRUE) au départ, la modification de la valeur de la commande de face-avant en fausse (FALSE) n'arrête pas l'exécution de cette boucle car la valeur n'est pas considérée en attendant que la boucle s'arrête et que le VI soit à nouveau exécuté. Si vous créez une boucle infinie sans le vouloir, vous pouvez l'arrêter en annulant le VI. Cliquez sur le bouton Abandonner pour arrêter l'exécution.

Auto-indexation

Les structures en boucle For et While peuvent automatiquement indexer et accumuler des tableaux se situant à leurs limites. Ces fonctions sont appelées l'*auto-indexation*. Lorsque vous reliez un tableau de source externe à un tunnel d'entrée sur une bordure de boucle et que vous activez l'auto-indexation sur le tunnel d'entrée, les composants de ce tableau entrent dans la boucle un par un, en commençant par le premier composant. La boucle indexe les éléments scalaires à partir des tableaux à une dimension, les tableaux à une dimension à partir des tableaux à deux dimensions, et ainsi de suite. L'action opposée se produit pour les tunnels de sortie – les éléments scalaires s'accumulent de manière séquentielle dans des tableaux à une dimension, les tableaux à une dimension s'accumulent dans des tableaux à deux dimensions, et ainsi de suite.

L'illustration suivante montre des tunnels sur les bordures d'une structure de boucle avec et sans auto-indexation. Le fil de liaison devient de plus en plus épais au fur et à mesure qu'il change de dimension sur la bordure de la boucle.



Vous utiliserez souvent des boucles For afin de traiter des tableaux de manière séquentielle. Par conséquent, les tunnels de tableau ont une auto-indexation activée par défaut lorsque vous reliez un tableau dans une boucle For ou en dehors. Les boucles While ne sont pas fréquemment utilisées dans ce but. L'auto-indexation n'est donc pas activée par défaut. Pour activer ou désactiver l'auto-indexation sur un tunnel se situant sur la bordure d'une boucle, vous devez ouvrir un menu local du tunnel et choisir **Activer l'indexation** ou **Désactiver l'indexation**.

Auto-indexation pour définir le nombre d'itérations de la boucle For

Lorsque vous commencez l'auto-indexation sur un tableau entrant une boucle For, elle définit automatiquement le nombre d'itérations en fonction de la taille du tableau. Par conséquent, elle élimine le besoin de liaison du terminal de comptage. Si vous commencez l'auto-indexation pour plusieurs tunnels ou si vous en définissez explicitement le nombre, ce dernier devient le plus petit des choix possibles. Par exemple, si deux tableaux auto-indexés entrent dans la boucle, avec respectivement 10 et 20 composants et que vous reliez une valeur de 15 au terminal de comptage, la boucle s'exécute dix fois et n'indexe que les dix premiers composants du deuxième tableau.

L'auto-indexation des tableaux de sortie reçoit un nouvel élément de sortie de chaque itération de la boucle. Par conséquent, les tableaux de sortie auto-indexés ont toujours une taille équivalente au nombre d'itérations (10 dans le précédent exemple).

Si l'auto-indexation est désactivée sur un tunnel de sortie, seule la valeur de l'élément de la dernière itération de la boucle sortira.

Auto-indexation des boucles While

Lorsque vous activez l'auto-indexation d'un tableau qui entre dans une boucle While, cette dernière indexe le tableau de la même manière qu'une boucle For. Toutefois, le nombre d'itérations qu'une boucle While exécute n'est pas limité par la taille du tableau car elle s'effectue tant qu'une certaine condition reste vraie (TRUE). Lorsqu'une boucle While s'indexe après la f... du tableau d'entrée, la valeur par défaut pour le type d'élément du tableau est passée dans la boucle. Les tableaux de sortie auto-indexés reçoivent un élément de sortie de chaque itération de la boucle While. Ils continuent à s'agrandir tant que la boucle While s'exécute.

**Remarque**

L'auto-indexation avec une boucle While qui itère un nombre de fois trop important peut provoquer une insuffisance de mémoire. Afin d'éviter ce genre de problèmes, si vous êtes un utilisateur de LabVIEW, consultez le chapitre 5, Tableaux, clusters et graphes, du Manuel de l'utilisateur LabVIEW pour obtenir plus d'informations concernant la conception des tableaux avec des boucles While.

Exécution d'une boucle For zéro fois

Lorsque vous définissez un nombre zéro, la boucle For n'exécute pas son sous-diagramme. La valeur de toutes les données scalaires qui quittent la boucle For respecte les règles suivantes.

- Un tableau de sortie créé par auto-indexation sur un tunnel de sortie est vide.
- La sortie d'un registre à décalage initialisé est la valeur initiale. Pour plus d'informations, consultez la section *Registres à décalage* de ce chapitre.
- Un tableau transmis par un tunnel de sortie non indexé est également vide.
- Tous les scalaires transmis par des tunnels de sortie non indexés ne sont pas définis et vous ne pouvez pas vous fier à leur valeur.

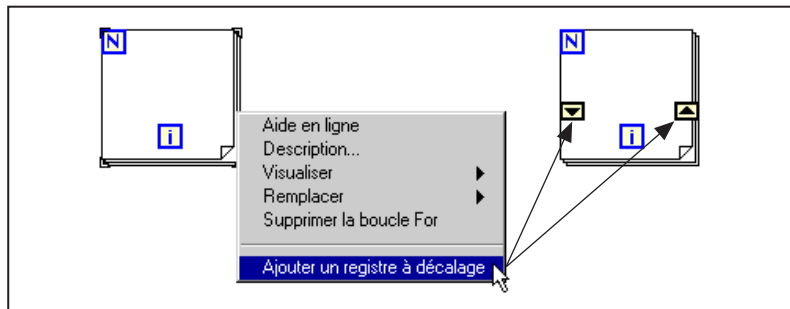
Le nombre de la boucle est défini sur zéro ou se met par défaut à zéro de deux manières. Vous pouvez soit auto-indexer un tableau d'entrée vide, soit explicitement relier un zéro ou un nombre négatif au terminal de comptage.

Lorsque vous auto-indexez des tableaux d'entrée ou définissez le nombre d'itérations de la boucle avec une variable, analysez le diagramme afin de déterminer si un nombre zéro survient, et le cas échéant, assurez-vous que le diagramme est capable de gérer un tableau vide.

Registres à décalage

Les registres à décalage auxquels vous avez accès par les boucles For et While sont des variables locales qui contre-réagissent ou transfèrent des valeurs de la fin d'une itération au début de l'itération suivante. En sélectionnant **Ajouter un registre à décalage** par le biais du menu local de la bordure de la boucle, vous pouvez créer un registre à n'importe quel endroit le long de la limite verticale, comme le montre l'illustration suivante. Cet élément de menu n'est pas disponible à partir des côtés

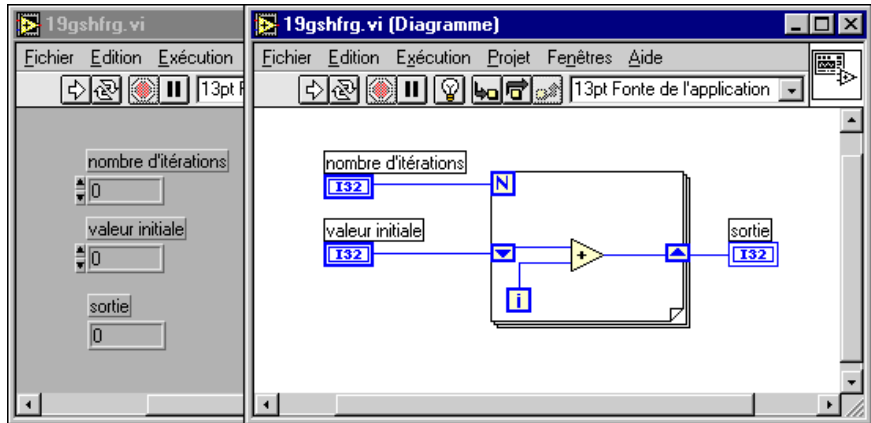
supérieurs et inférieurs de la structure. Vous pouvez repositionner un registre à décalage le long de la limite verticale en le faisant glisser.




Le registre à décalage possède deux terminaux directement opposés l'un à l'autre sur les côtés verticaux de la bordure de la boucle. Le *terminal de droite*, le rectangle avec la flèche vers le haut, stocke les données à la fin de l'exécution de l'itération. Le G décale ces données à la fin de l'itération et les transmet au *terminal de gauche*, le rectangle avec la flèche vers le bas, au moment de l'itération suivante. Vous pouvez utiliser les registres à décalage pour n'importe quel type de données, mais les données que vous reliez aux terminaux de chaque registre doivent être du même type. Vous pouvez créer plusieurs registres à décalage sur une structure donnée.

Pour initialiser un registre à décalage, reliez une valeur de l'extérieur de la boucle au terminal de gauche. Si vous n'initialisez pas le registre, la boucle utilise la valeur saisie dans le registre lors de la dernière exécution de la boucle ou la valeur par défaut pour son type de données si la boucle n'a jamais été exécutée. Utilisez les registres à décalage initialisés pour assurer un comportement cohérent. Les utilisateurs de LabVIEW peuvent se référer au chapitre 3, *Boucles et graphes déroulants*, du *Manuel de l'utilisateur LabVIEW* pour obtenir plus d'informations sur l'utilisation des registres à décalage non-initialisés. Les utilisateurs de BridgeVIEW peuvent consulter le chapitre 10, *Loops and Charts*, du *BridgeVIEW User Manual*.

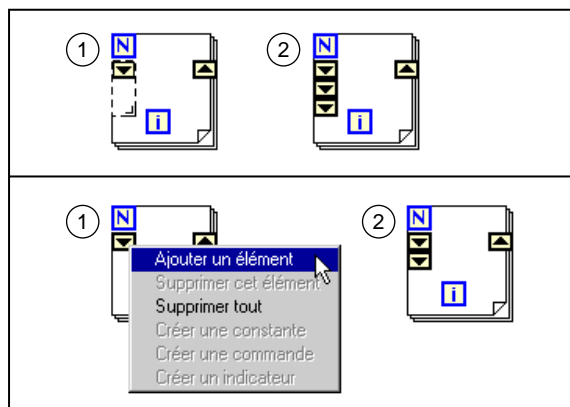
Lorsque la boucle termine son exécution, la dernière valeur stockée dans le registre à décalage est conservée dans le terminal de droite. Si ce dernier est relié à l'extérieur de la boucle, cette valeur est transmise en dehors de la boucle dès que son exécution se termine. L'illustration suivante montre cette propriété.



La partie gauche du registre à décalage peut contenir plusieurs terminaux. Elle peut donc stocker plusieurs valeurs précédentes.


 Curseur de redimensionnement

Pour ajouter ou supprimer des terminaux d'un registre à décalage donné, utilisez le curseur de redimensionnement pour ajuster la taille des terminaux de gauche. Autrement, vous pouvez utiliser la commande **Ajouter un élément** du menu local du registre à décalage afin d'ajouter au registre à décalage des terminaux supplémentaires à gauche. Les terminaux ajoutés s'affichent directement en dessous de celui sur lequel vous avez ouvert un menu local. Utilisez la commande **Supprimer un élément** pour supprimer le terminal sur lequel vous avez ouvert un menu local. L'illustration suivante montre les deux méthodes. La seule manière de supprimer des terminaux reliés supplémentaires consiste à choisir **Supprimer un élément** dans le menu local du registre à décalage. Si vous sélectionnez **Supprimer tout**, vous supprimez le registre à décalage.



Le terminal en haut à gauche conserve la valeur de la précédente itération, $i - 1$. Le terminal qui se situe juste en dessous contient la valeur de l'itération $i - 2$, et ainsi de suite pour tous les autres terminaux.

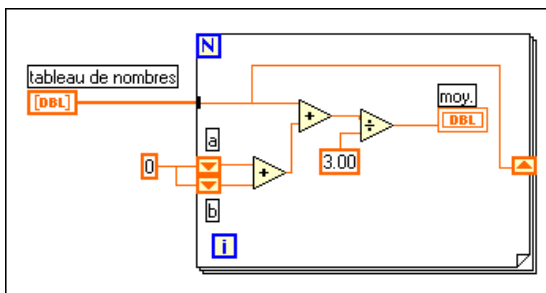
Si vous initialisez un terminal de gauche d'un registre à décalage, vous devez également initialiser tous les terminaux supplémentaires.

Le pseudo-code suivant montre une routine moyenne d'exécution à trois variables équivalente au diagramme G.

```

a = b = 0
for i = 0 to N - 1
    avg = (x[i] + a + b)/3
    b = a
    a = x[i]

```

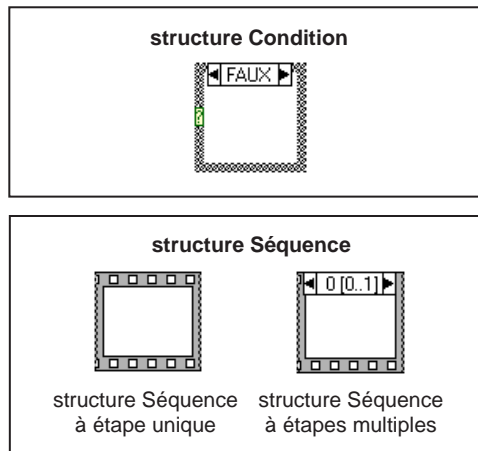


Pour obtenir un exemple de paramétrage d'un registre à décalage, consultez la bibliothèque `examples\general\structs.llb\Random Average.vi`.

Structures Condition et Séquence

Les structures Condition et les structures Séquence peuvent toutes deux posséder plusieurs sous-diagrammes, configurés comme un jeu de cartes, dont un seul est visible à la fois. En haut de chaque bordure de structure se trouve la *fenêtre d'affichage du sous-diagramme* qui contient un *identificateur de diagramme* au centre et des boutons de décrémentation et d'incrémentement de chaque côté. En ce qui concerne la structure Condition, l'identificateur de diagramme affiche les valeurs générant l'exécution du sous-diagramme actuellement affiché. Quant à la structure Séquence, l'identificateur de diagramme indique le sous-diagramme actuellement affiché. Si l'identificateur de diagramme est numérique, il est suivi par une gamme d'identificateurs de diagramme, qui montre les

valeurs minimales et maximales de la structure contenant les sous-diagrammes.



En cliquant sur le bouton de décrémentation (à gauche) ou d'incrémement (à droite), vous affichez respectivement le sous-diagramme précédent ou suivant. La décrémentation à partir du premier sous-diagramme affiche le dernier et l'incrémement à partir du dernier sous-diagramme affiche le premier. D'autres utilisations de la fenêtre d'affichage vous sont expliquées dans les sections *Structures Condition* et *Structures Séquence* de ce chapitre.

Consultez le VI `examples\general\structs.llb\SquareRoot.vi` pour obtenir un exemple de structure Condition.

Structures Condition

Les structures Condition sélectionnent un sous-diagramme unique, ou une condition, à exécuter selon une valeur d'entrée appelée sélecteur. Avec les structures Condition, vous pouvez effectuer les opérations suivantes :

- Spécifier des listes et des gammes de valeurs de sélecteur qui correspondent à une condition.
- Utiliser un entier, un booléen, une chaîne de caractères ou un type énuméré comme sélecteur.
- Spécifier une condition par défaut (ou une action).
- Trier les conditions en fonction de la première valeur du sélecteur.

Lorsque vous placez une structure Condition sur un diagramme, vous entrez les valeurs du sélecteur directement dans l'étiquette de sélection de

la structure Condition. Vous pouvez également éditer les valeurs du sélecteur en utilisant l'outil Texte. Vous pouvez spécifier une seule valeur ou des listes et des gammes de valeurs qui sélectionnent la condition. Pour indiquer une liste, séparez les valeurs par des virgules, de la manière suivante : 1, 0, 5, 10. Une gamme est spécifiée par 10..20, ce qui signifie que tous les nombres entre 10 et 20 sont inclus. Vous pouvez également utiliser des gammes indéterminées telles que ..0 (tous les nombres inférieurs ou égaux à 0) ou 100.. (tous les nombres supérieurs ou égaux à 100). Les listes et les gammes peuvent être combinées, telles que ..5, 7..10, 12, 13, 14. Lorsque vous entrez un sélecteur qui contient des gammes qui se chevauchent, la structure Condition affiche à nouveau le sélecteur dans une forme plus compacte. Le précédent exemple est réaffiché sous la forme ..10, 12..14.

Vous pouvez également utiliser des valeurs chaîne de caractères et énumérées dans le sélecteur de condition. Ces dernières s'affichent toujours entre des guillemets, de la manière suivante : "rouge", "vert", "bleu". Toutefois, vous n'avez pas besoin de taper entre les guillemets lorsque vous entrez des valeurs à moins que la chaîne de caractères ou l'énumérée contienne une virgule ou le symbole "..".



Remarque

Si vous entrez une valeur de sélecteur qui n'est pas du même type que l'objet relié à l'icône du sélecteur, la valeur s'affiche alors en rouge et votre VI ne peut pas être exécuté. De même, étant donné l'erreur d'arrondi possible inhérente à l'arithmétique de la virgule flottante, LabVIEW n'autorise pas l'utilisation des nombres à virgule flottante dans les étiquettes du sélecteur de condition. Si vous le faites, le type est arrondi à l'entier le plus proche comme dans les précédentes versions de LabVIEW. Si vous essayez d'entrer une valeur à virgule flottante telle que 1,2 dans le sélecteur de Condition, elle s'affiche en rouge.

La Condition présentée ci-après est de type numérique. Par conséquent, la valeur chaîne de caractères du sélecteur s'affiche en rouge.

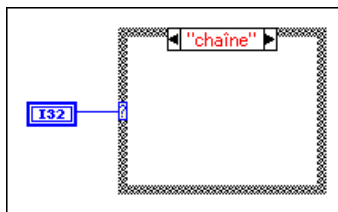


Figure 19-1. Incohérence du type de la structure Condition

Pour ajouter des conditions supplémentaires, suivez la procédure proposée dans la section *Ajout de sous-diagrammes* de ce chapitre.



Remarque

Les instructions de condition des autres langages de programmation n'exécutent généralement pas une condition si une valeur de condition est hors gamme. Dans le G, vous devez inclure une condition par défaut qui gère les valeurs hors gamme, ou lister explicitement toutes les valeurs d'entrée possibles.

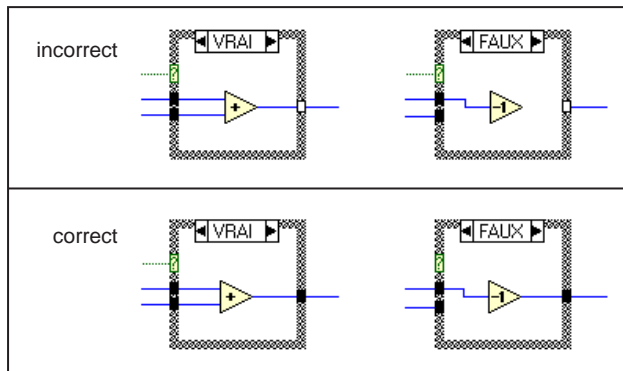
Vous pouvez positionner le sélecteur à n'importe quel endroit le long de la bordure gauche et vous devez le câbler. Le sélecteur s'adapte automatiquement au type de données. Si vous changez une valeur numérique reliée au sélecteur par une valeur booléenne, les conditions 0 et 1 changent en faux (FALSE) et vrai (TRUE). Si d'autres conditions existent (de 2 à n), le G ne les détruit pas, dans le cas où la modification du type de données est accidentelle. Toutefois, vous devez supprimer ces conditions pour que la structure puisse s'exécuter.

Le même principe s'applique si vous reliez une énumération au sélecteur et qu'il y a plus de conditions que d'éléments dans l'énumération. L'identificateur de diagramme de telles conditions s'affiche en identificateur numérique rouge pour signaler que ces conditions doivent être supprimées avant l'exécution de la structure.

Lorsque vous convertissez des valeurs de sélecteur de condition en un autre type, prenez en considération le fait suivant. Si vous convertissez des valeurs telles que 23 en une chaîne de caractères, la valeur de la chaîne de caractères est 23. Si vous convertissez une chaîne de caractères en une valeur numérique, seules les chaînes de caractères de sélecteur qui représentent un nombre sont converties en valeurs numériques. Les autres valeurs restent des chaînes de caractères. Si vous convertissez un nombre en un booléen, 0 est converti en FALSE et 1 en TRUE. Les autres valeurs deviennent par contre des chaînes de caractères.

Les données de tous les terminaux d'entrée (tunnels et terminal de sélection) sont disponibles pour toutes les conditions. Les conditions ne doivent pas nécessairement recevoir des données d'entrée ou fournir des données de sortie, mais *si une des conditions fournit des données de sortie, elles doivent toutes le faire*. Si vous ne reliez pas les données à un tunnel de sortie à partir de chaque condition, le tunnel devient blanc, comme le montre l'exemple du haut de l'illustration suivante et le bouton d'**exécution** affiche une flèche brisée.

Lorsque toutes les conditions fournissent des données au tunnel, il devient noir, comme le montre l'exemple du bas, et le bouton d'exécution affiche une flèche en bon état.



Les éléments du menu local de la structure Condition comprennent des éléments pour les conditions non booléennes, comme le montre l'illustration suivante.

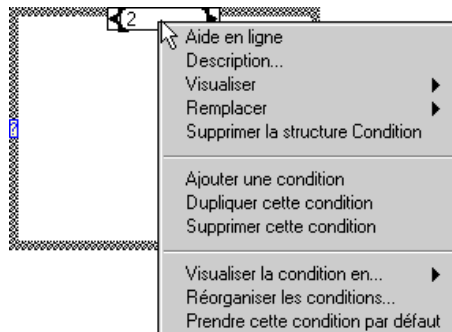


Figure 19-2. Eléments du menu local d'une structure Condition

La fonction “Ajouter une condition” rajoute une condition à la suite de la condition visible. Vous ne pouvez pas ajouter de condition avant la condition visible. Vous pouvez modifier l'ordre des conditions en sélectionnant l'élément **Réorganiser les conditions...** qui affiche la boîte de dialogue suivante.

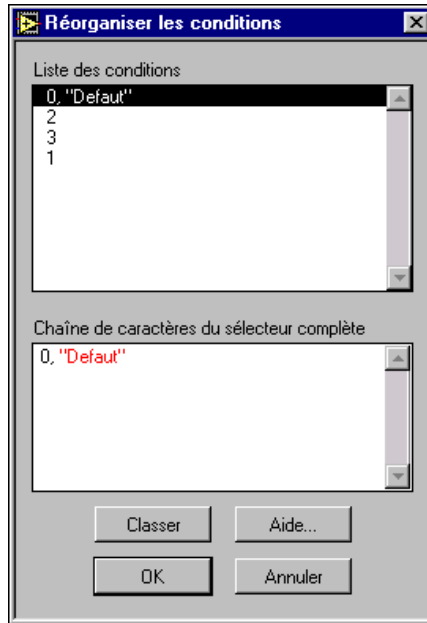


Figure 19-3. Boîte de dialogue “Réorganiser les conditions”

Le bouton **Classer** classe les valeurs de sélecteur de condition en fonction de la première valeur de sélecteur. Pour modifier l’emplacement d’un sélecteur, cliquez sur la valeur du sélecteur que vous voulez déplacer et faites-le glisser vers le nouvel emplacement. **Toute la chaîne de caractères du sélecteur** affiche le sélecteur de condition sélectionné s’il est trop long pour être affiché dans la **Liste des conditions**. Une aide contextuelle sur toutes les commandes est accessible dans cette boîte de dialogue par un clic sur le bouton **Aide**.

 **Remarque**

Cette fonction ne modifie que l’ordre dans lequel les conditions apparaissent dans la structure Condition. Elle n’affecte pas le comportement de l’exécution de la structure Condition.

L'élément du menu local **Changer cette étape en défaut** spécifie si une condition donnée s'exécute si la valeur du sélecteur n'est pas listée dans la structure Condition. Le mot **défaut** est listé dans la valeur du sélecteur en haut de la structure Condition. Vous devez spécifier une condition par défaut pour une structure Condition si elle ne contient pas de sélecteurs pour chaque valeur de sélecteur possible.

L'ajout, le déplacement et la suppression de sous-diagrammes de Condition sont traités à la suite de la section suivante sur les structures Séquence, car ces opérations sont identiques aux deux structures.

Structures Séquence

La structure Séquence, qui ressemble à un film, comporte un ou plusieurs sous-diagrammes, ou *cadres*, qui s'exécutent de manière séquentielle. Pour obtenir un exemple de VI utilisant une structure Séquence, consultez la bibliothèque `examples\general\structs.llb\TimingTemplate.vi`.

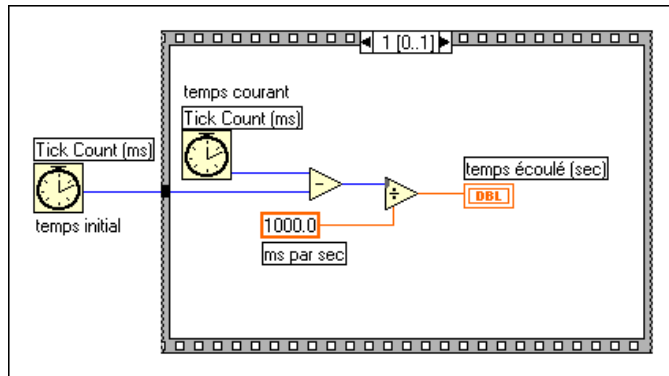
La détermination de l'ordre d'exécution d'un programme par l'arrangement de ses éléments en séquence s'appelle la *programmation séquentielle*. Le BASIC, le C et la plupart des autres langages de programmation possèdent une programmation séquentielle inhérente. En effet, les instructions s'exécutent dans l'ordre d'apparition dans le programme. La structure Séquence est une manière d'obtenir une programmation séquentielle lorsque la dépendance des données ne suffit pas. Une structure Séquence exécute le cadre 0, suivi par le cadre 1, puis le cadre 2, et ce jusqu'au dernier cadre. Les données ne quittent la structure qu'après l'exécution du dernier cadre.

A l'intérieur de chaque cadre, comme dans le reste du diagramme, la dépendance des données détermine l'ordre d'exécution des nœuds.

Vous utiliserez la structure Séquence afin de contrôler l'ordre d'exécution des nœuds qui ne dépendent pas de données. Un nœud qui reçoit ses données directement ou indirectement d'un autre nœud a une dépendance de données sur l'autre nœud et s'exécute toujours à la fin de l'autre nœud. Il n'est pas nécessaire d'utiliser la structure Séquence lorsque la dépendance des données existe ou lorsque l'ordre d'exécution n'a pas d'importance.

Cette situation a souvent lieu lorsque vous voulez déterminer le temps pris par une fonction. La fonction "Compteur d'impulsions d'horloge" renvoie le temps en cours en millisecondes. Le temps n'est pas nécessaire à l'exécution de la fonction. Il n'y a donc pas de dépendance de données entre la mesure du temps et la fonction. L'illustration suivante montre la

fonction “Compteur d’impulsions d’horloge” à gauche de la structure Séquence qui s’exécute avant cette dernière. Ensuite, la fonction s’exécute dans le cadre 0 et le calcul du temps écoulé est effectué dans le cadre 1. La structure Séquence assure le bon ordre d’exécution (le cadre 0 n’est pas illustré).

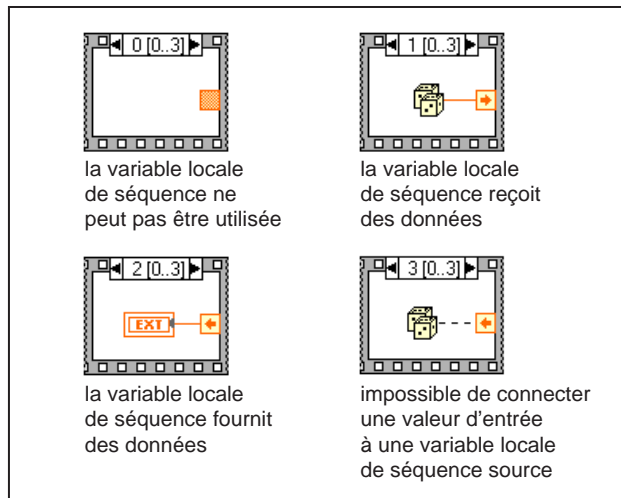


Les tunnels de sortie des structures Séquence peuvent n’avoir qu’une seule source de données. Ce n’est pas le cas des structures Condition. La sortie peut être émise à partir de n’importe quel cadre, mais souvenez-vous que les données ne quittent la structure que lorsqu’elle termine son exécution dans sa totalité et non pas à la fin de chaque cadre. Les données des tunnels d’entrée sont accessibles à tous les cadres, comme dans les structures Condition.

Pour passer des données d’un cadre au suivant, utilisez un terminal appelé *variable locale de séquence*. Pour obtenir une variable locale de séquence, choisissez **Ajouter une variable locale de séquence** dans le menu local de la bordure de la structure. Cet élément n’est pas disponible si vous ouvrez un menu local sur une variable locale de séquence ou sur une fenêtre d’affichage de sous-diagramme. Vous pouvez faire glisser le terminal vers un emplacement inoccupé sur la bordure. Utilisez la commande **Supprimer** dans le menu local de la variable locale de séquence pour supprimer un terminal.

Une flèche pointant vers l’extérieur s’affiche dans le terminal de la variable locale de séquence du cadre qui contient la source de données. Le terminal des cadres suivants contient une flèche pointant vers l’intérieur, ce qui signifie que le terminal est une source pour ce cadre. Vous ne pouvez pas utiliser la variable locale de séquence dans les cadres qui précèdent le cadre source et par conséquent elle s’affiche sous forme de rectangle grisé.

Les illustrations suivantes montrent le terminal de la variable locale de séquence.

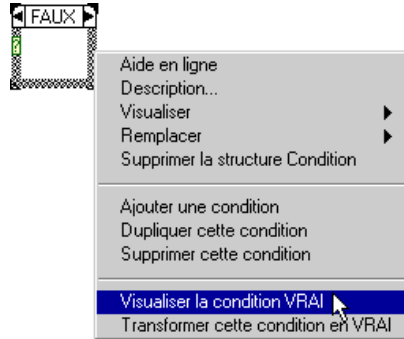


Edition des structures Séquence et Condition

Etant donné que l'édition et la manipulation des structures Séquence et Condition impliquent des techniques similaires, les exemples suivants ne montrent qu'une structure Condition et ses menus locaux. En ce qui concerne la structure Séquence, remplacez le mot *condition* par *étape*. Une nouvelle structure Condition possède deux conditions mais peut être éditée pour n'avoir qu'une seule condition. Une nouvelle structure Séquence possède une seule étape.

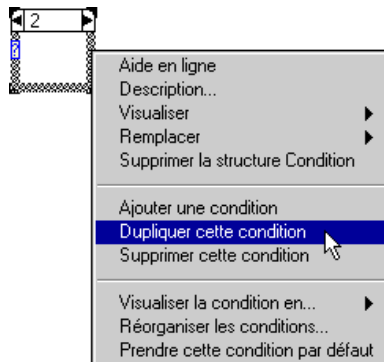
Déplacement entre les sous-diagrammes

La manière la plus rapide de visualiser le sous-diagramme inférieur ou supérieur suivant consiste à cliquer sur le bouton de décrémentation ou d'incrémement de la fenêtre d'affichage. Si vous voulez sauter plusieurs sous-diagrammes, cliquez sur l'identificateur de sous-diagramme et sélectionnez le sous-diagramme de destination dans le menu local, comme le montre l'illustration suivante. Vous pouvez également utiliser la commande **Visualiser la condition en...** dans le menu local de la bordure.



Ajout de sous-diagrammes

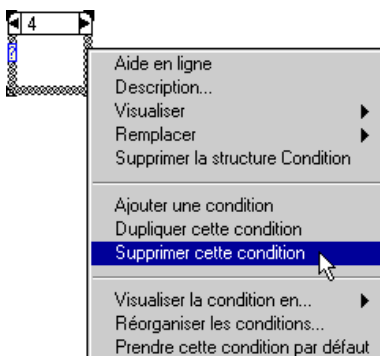
Si vous sélectionnez **Dupliquer cette condition** soit dans le menu local de la bordure de la structure soit dans celui de l'identificateur de diagramme, comme indiqué ci-dessous, une copie du sous-diagramme visible est insérée à sa suite.



Lorsque vous ajoutez ou supprimez des sous-diagrammes d'une structure Séquence, l'éditeur ajuste automatiquement les identificateurs de diagrammes conformément aux sous-diagrammes insérés ou supprimés. En ce qui concerne les structures Condition, l'ordre des sous-diagrammes n'affecte pas l'exécution du programme, mais il peut être modifié en utilisant l'élément **Réorganiser les conditions...** du menu local.

Suppression de sous-diagrammes

Pour supprimer le sous-diagramme visible, choisissez la commande **Supprimer cette condition** dans le menu local de la bordure de la structure, illustrée ci-dessous. Elle n'est pas disponible s'il n'y a qu'un seul sous-diagramme.

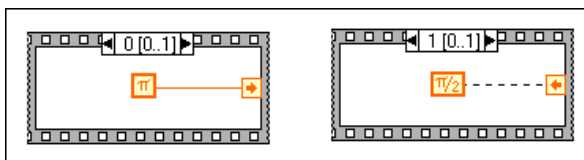


Problèmes de câblage de structure

Les sections suivantes traitent des erreurs de connexion des structures.

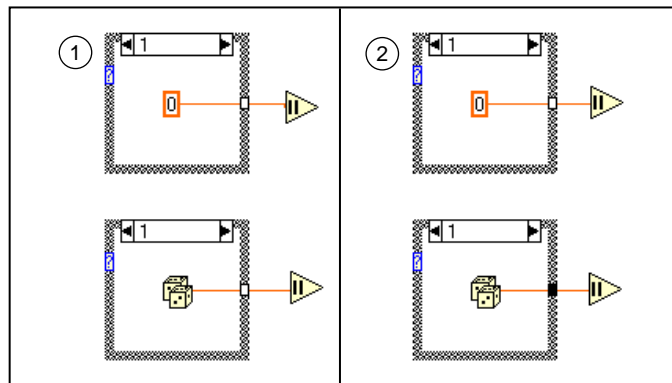
Affectation de plusieurs valeurs à une variable locale de séquence

Vous pouvez affecter une valeur à la variable locale d'une structure Séquence dans un seul cadre, même si vous pouvez utiliser la valeur dans tous les cadres suivants. L'illustration ci-après à gauche montre la valeur π affectée à la variable locale de séquence dans le cadre 0. Si vous essayez d'affecter une autre valeur à la même variable locale dans le cadre 1, vous créez une liaison incorrecte. Cette erreur constitue une variation de l'erreur de sources multiples.



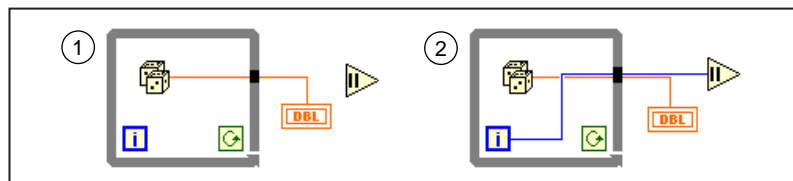
Tunnel non câblé dans toutes les conditions d'une structure Condition

Le câblage d'une structure Condition à un objet qui se situe à l'extérieur de la structure crée un tunnel incorrect si vous ne le reliez pas à une source dans toutes les conditions, comme vous le montre la première partie de l'exemple suivant. Cette erreur constitue une variation de l'erreur d'absence de source car il y a au moins une condition qui ne produit pas de valeur de données à l'exécution. Le câblage du tunnel à toutes les conditions, comme vous le montre la deuxième partie de cet exemple, corrige le problème. Ce n'est pas une erreur de sources multiples car une seule condition s'exécute et elle ne produit qu'une valeur de sortie par exécution de la structure Condition.

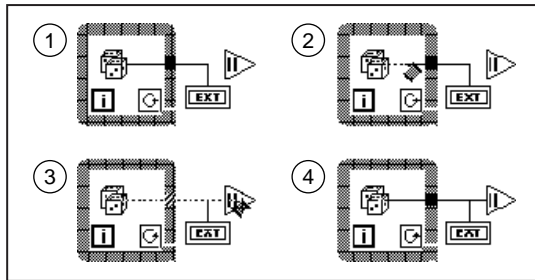


Chevauchement de tunnels

Etant donné que le G crée des tunnels au fur et à mesure de votre câblage, les tunnels finissent parfois par se chevaucher. Le chevauchement des tunnels n'affecte pas l'exécution du diagramme mais peut en compliquer l'édition. Evitez de créer des tunnels qui se chevauchent. Si toutefois cela se produit, faites glisser un des tunnels pour les séparer. Observez l'exemple suivant.



Il est difficile de déterminer quel tunnel se trouve au-dessus de l'autre. Vous pouvez faire des erreurs si vous essayez d'en relier un des deux alors qu'ils se chevauchent. Dans ce cas, supprimez les fils incorrects et réessayez.

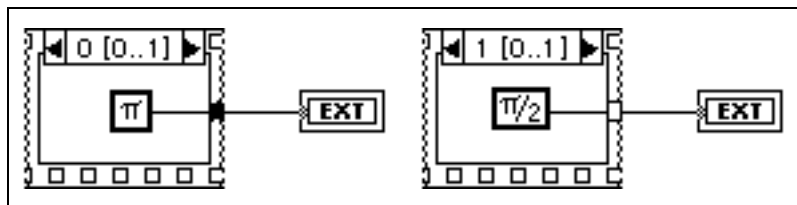


Si vous devez relier un objet à l'intérieur d'une structure à un objet à l'extérieur alors qu'un tel fil de liaison existe déjà, *ne les reliez pas* de nouveau à travers la structure comme dans la précédente illustration. Commencez le deuxième fil de liaison au niveau du tunnel. Dans cet exemple, deux tunnels qui se chevauchent ne provoquent pas de problèmes. Mais si c'était une structure Condition, deux tunnels incorrects qui se chevauchent pourraient sembler être reliés à chaque condition. Vous pouvez toujours supprimer tous les fils de liaison d'un tunnel pour le faire disparaître et effectuer ensuite un câblage correct.

Si vos tunnels de VI se chevauchent entièrement, un avertissement s'affiche dans la fenêtre "Liste des erreurs" si vous sélectionnez **Visualiser les mises en garde**. Pour plus d'informations sur ces problèmes, consultez la section *Comprendre les mises en garde* du chapitre 4, *Exécution et mise au point des VIs et sous-VIs*.

Câblage à partir de plusieurs étapes d'une structure Séquence

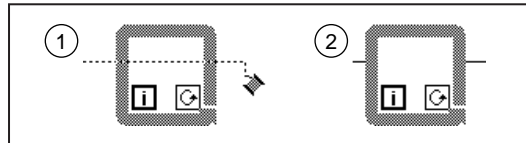
L'illustration suivante montre un autre type d'erreur de sources multiples. Deux étapes de structure Séquence essaient d'affecter des valeurs au même tunnel. Le tunnel devient blanc pour signaler l'erreur.



Câblage par-dessous plutôt qu'à travers une structure

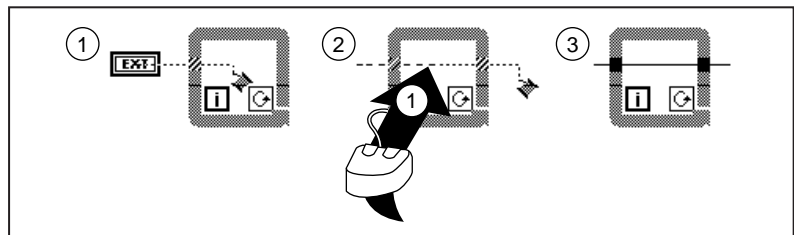
Pour câbler à travers une structure, vous devez soit cliquer à l'intérieur soit sur la bordure de la structure, comme vous le montre l'illustration suivante.

Si vous ne cliquez pas à ces endroits, le fil de liaison passe par-dessous la structure.



Lorsque l'outil Bobine traverse la bordure gauche de la structure, un tunnel mis en surbrillance s'affiche pour indiquer que l'éditeur créera un tunnel à cet emplacement dès que vous appuierez sur le bouton de la souris. Si vous continuez à faire glisser l'outil Bobine à travers la structure sans cliquer avant d'avoir atteint la bordure droite, un deuxième tunnel mis en surbrillance s'affiche sur la bordure droite. Si vous continuez à faire glisser l'outil Bobine au-delà de la bordure droite sans cliquer, les deux tunnels disparaissent et le fil de liaison passe sous la structure plutôt qu'à travers.

Observez l'illustration suivante.



Toutefois, si vous fixez le fil de liaison à l'intérieur de la structure, il traversera la structure même si vous continuez à faire glisser l'outil Bobine au-delà de la bordure droite.

Suppression de structures sans effacer les éléments qui se trouvent à l'intérieur

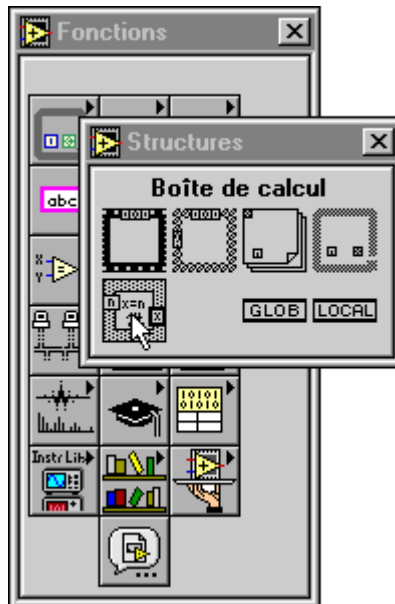
Vous pouvez supprimer une structure (boucle For ou While, structure Condition ou Séquence) sans perdre le contenu de la structure. Si vous ouvrez un menu local sur un de ces objets, vous verrez un élément permettant d'effectuer la suppression de la structure. Dans le cas des boucles For et While, le contenu de la boucle est copié dans le diagramme de base. De plus, tous les fils de liaison connectés par des tunnels sont reliés automatiquement les uns aux autres.

En ce qui concerne la structure Condition ou la structure Séquence, la suppression de la structure ne conserve que le cadre ou la condition en cours. Tous les autres cadres ou conditions sont supprimés. La perte des cadres ou des conditions masqués vous est signalée et vous pourrez à ce moment annuler l'opération.

Boîtes de calcul

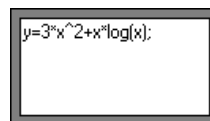


Ce chapitre présente la manière d'utiliser la Boîte de calcul pour exécuter des formules mathématiques sur le diagramme. La Boîte de calcul est accessible dans la palette **Fonctions»Structures**.



Utilisation des Boîtes de calcul

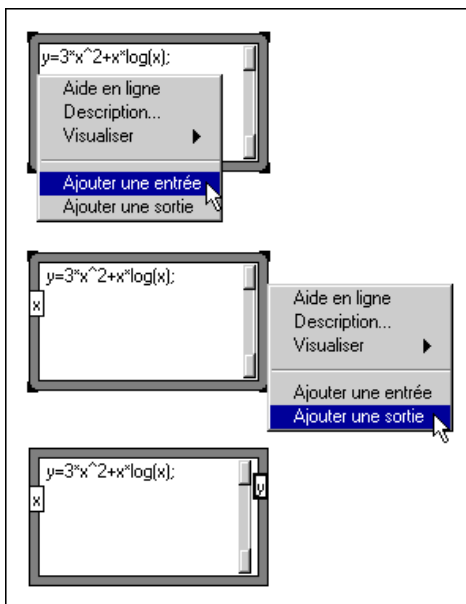
La Boîte de calcul est une boîte redimensionnable identique aux quatre structures (boucles For et While, structures Condition et Séquence). Toutefois, au lieu de contenir un sous-diagramme, la Boîte de calcul contient une ou plusieurs instructions exprimées sous forme de formules et délimitées par un point-virgule, comme dans l'exemple suivant.



Les instructions de formules utilisent une syntaxe identique à la plupart des langages de programmation basés sur du texte en ce qui concerne les expressions arithmétiques. Vous pouvez ajouter des commentaires en les mettant entre des barres obliques-astérisques (*/*commentaire*/*).

Pour obtenir un exemple qui utilise une Boîte de calcul, consultez la bibliothèque `examples\general\structs.llb\Equations.vi`.

Le menu local de la bordure de la Boîte de calcul contient des éléments permettant d'ajouter des variables d'entrée et de sortie, comme le montre l'illustration suivante.

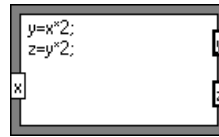


Les variables de sortie possèdent une bordure plus épaisse.

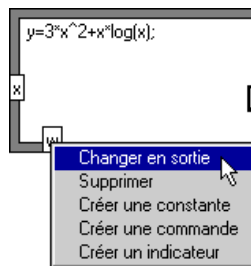
Le nombre de variables ou de formules dans une Boîte de calcul n'est pas limité. Deux entrées ou deux sorties ne peuvent pas être nommées de manière identique. Toutefois, une sortie peut avoir le même nom qu'une entrée.

Chaque variable utilisée dans la Boîte de calcul doit être déclarée comme entrée ou sortie. Les variables intermédiaires, c'est-à-dire toute variable associée au résultat d'une opération calculée à la suite d'une ou de plusieurs entrées dans le nœud mais avant la sortie finale du nœud, doit être déclarée comme sortie. Toutefois, il n'est pas nécessaire que les variables intermédiaires soient reliées aux nœuds externes de la Boîte de calcul.

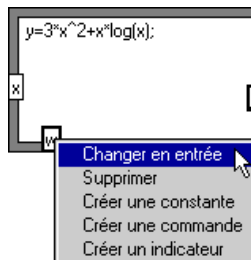
Dans l'illustration suivante, la variable y et la variable z doivent toutes deux être déclarées comme des sorties, même si la valeur y n'est pas reliée aux nœuds externes.



Vous pouvez changer une entrée en sortie en sélectionnant **Changer en sortie** dans le menu local, comme le montre l'illustration suivante.



Vous pouvez changer une sortie en entrée en sélectionnant **Changer en entrée** dans le menu local, comme indiqué ci-dessous.



Toutes les variables sont des scalaires numériques à virgule flottante dont la précision dépend de la configuration de votre ordinateur. Les variables ne peuvent pas avoir d'unités. Toutes les variables d'entrée qui s'affichent dans les formules doivent être câblées. Au moins une instruction doit être affectée à toutes les variables de sortie câblées. Les variables de sortie doivent être situées à la gauche du signe égal (=). Une variable de sortie peut apparaître dans une expression à droite du signe égal (=), mais le G ne vérifie pas si elle est affectée à une précédente instruction. Lorsqu'une

affectation est une sous-expression, la valeur de la sous-expression est la valeur affectée. Par exemple :

$$x = \sin (y = \pi/3)$$

affecte $\frac{\pi}{3}$ à y , et affecte ensuite $\sin\left(\frac{\pi}{3}\right)$ à x .

Si une erreur de syntaxe se produit, vous pouvez cliquer sur le bouton **Flèche brisée** pour voir la liste des erreurs. La Boîte de calcul affiche une partie de la formule dans la liste avec un symbole # qui marque l'emplacement auquel la boîte de formule a détecté l'erreur.

Fonctions et opérateurs de la Boîte de calcul

Tous les noms de fonctions doivent être en caractères minuscules. Le tableau 20-1 montre les noms des fonctions de la Boîte de calcul.

Tableau 20-1. Fonctions de la Boîte de calcul

Fonction	Nom de la fonction G correspondante	Description
abs(x)	Valeur absolue	Retourne la valeur absolue de x .
acos(x)	Arc cosinus	Calcule l'arc cosinus de x en radians.
acosh(x)	Arc cosinus hyperbolique	Calcule l'arc cosinus hyperbolique de x en radians.
asin(x)	Arc sinus	Calcule l'arc sinus de x en radians.
asinh(x)	Arc sinus hyperbolique	Calcule l'arc cosinus hyperbolique de x en radians.
atan(x)	Arc tangente	Calcule l'arc tangente de x en radians.
atanh(x)	Arc tangente hyperbolique	Calcule l'arc tangente hyperbolique de x en radians.

Tableau 20-1. Fonctions de la Boîte de calcul (Suite)

Fonction	Nom de la fonction G correspondante	Description
$\text{ceil}(x)$	Arrondir à +Infini	Arrondit x à l'entier supérieur suivant (le plus petit entier supérieur ou égal à x).
$\text{cos}(x)$	Cosinus	Calcule le cosinus de x en radians.
$\text{cosh}(x)$	Cosinus hyperbolique	Calcule le cosinus hyperbolique de x en radians.
$\text{cot}(x)$	Cotangente	Calcule la cotangente de x en radians ($1/\tan(x)$).
$\text{csc}(x)$	Cosécante	Calcule la cosécante de x en radians ($1/\sin(x)$).
$\text{exp}(x)$	Exponentiel	Calcule la valeur de e élevée à la puissance x .
$\text{expm1}(x)$	Exponentiel(Arg) – 1	Calcule la valeur de e élevée à la puissance x moins un ($e^x - 1$).
$\text{floor}(x)$	Arrondir à –Infini	Tronque x à l'entier inférieur suivant (le plus grand entier inférieur ou égal à x).
$\text{getexp}(x)$	Mantisse & Exposant	Retourne l'exposant de x .
$\text{getman}(x)$	Mantisse & Exposant	Retourne la mantisse de x .
$\text{int}(x)$	Arrondir à l'entier le plus proche	Arrondit son argument à l'entier le plus proche.
$\text{intrz}(x)$	Arrondir à 0	Arrondit x à l'entier le plus proche entre x et zéro.
$\text{ln}(x)$	Logarithme naturel	Calcule le logarithme naturel de x (base e).

Tableau 20-1. Fonctions de la Boîte de calcul (Suite)

Fonction	Nom de la fonction G correspondante	Description
$\lnp1(x)$	Logarithme naturel (Arg + 1)	Calcule le logarithme naturel de $(x + 1)$.
$\log(x)$	Logarithme de base 10	Calcule le logarithme de x (base 10).
$\log2(x)$	Logarithme de base 2	Calcule the logarithme de x (base 2).
$\max(x,y)$	Max & Min	Compare x et y et retourne la plus grande valeur.
$\min(x,y)$	Max & Min	Compare x et y et retourne la plus petite valeur.
$\text{mod}(x,y)$	Quotient & Reste	Calcule le reste de x/y lorsque le quotient est arrondi à moins l'infini.
$\text{rand}()$	Nombre aléatoire (0–1)	Produit un nombre à virgule flottante compris exclusivement entre 0 et 1.
$\text{rem}(x,y)$	Reste	Calcule le reste de x/y lorsque le quotient est arrondi à l'entier le plus proche.
$\text{sec}(x)$	Sécante	Calcule la sécante de x radians ($1/\cos(x)$).
$\text{sign}(x)$	Signe	Retourne 1 si x est supérieur à 0, retourne 0 si x est égal à 0, et retourne -1 si x est inférieur à 0.
$\text{sin}(x)$	Sinus	Calcule le sinus de x radians.
$\text{sinc}(x)$	Sinc	Calcule le sinus de x divisé par x radians ($\text{sin}(x)/x$).
$\text{sinh}(x)$	Sinus hyperbolique	Calcule le sinus hyperbolique de x en radians.

Tableau 20-1. Fonctions de la Boîte de calcul (Suite)

Fonction	Nom de la fonction G correspondante	Description
$\text{sqrt}(x)$	Racine carrée	Calcule la racine carrée de x .
$\text{tan}(x)$	Tangente	Calcule la tangente de x en radians.
$\text{tanh}(x)$	Tangente hyperbolique	Calcule la tangente hyperbolique de x en radians.
x^y	x^y	Calcule la valeur de x élevée à la puissance y .

Syntaxe de la Boîte de calcul

La syntaxe de la Boîte de calcul est résumée ci-dessous en utilisant la notation Backus–Naur Form (BNF). Les crochets enferment des éléments optionnels.

```

<assignlst> := <outputvar> = <aexpr> ; [ <assignlst> ]
<aexpr> := <expr> | <outputvar> = <aexpr>
<expr> := <expr> <binaryoperator> <expr>
        | <unaryoperator> <expr>
        | <expr> ? <expr> : <expr>
        | ( <expr> )
        | <inputvar>
        | <outputvar>
        | <const>
        | <function> ( <arglist> )
<binaryoperator> := + | - | * | / | ^ | != | ==
        | > | < | >= | <= | && | ||
<unaryoperator> := + | - | !
<arglist> := <aexpr> [ , <arglist> ]
<const> := pi | <number>

```

La priorité des opérateurs est la suivante, de la plus petite à la plus grande.

=	affectation
? :	conditionnel
	logique ou
&&	logique et
!= ==	inégalité, égalité
< > <= >=	autre relation : inférieur à, supérieur à, inférieur ou égal à, supérieur ou égal à
+ -	addition, soustraction
* /	multiplication, division
+ - !	unitaire : positif, négatif, non logique
^	élévation à une puissance

Les opérateurs d'élévation à une puissance et d'affectation sont associatifs à droite. Tous les autres opérateurs binaires sont associatifs à gauche. La valeur numérique de vraie (TRUE) est 1 et celle de fausse (FALSE) est 0 (pour la sortie). La valeur logique de 0 est fausse (FALSE), et celle de tout nombre différent de 0 est vraie (TRUE). La valeur logique de l'expression conditionnelle

```
<lexpr> ? <texpr>: <fexpr>
```

est <texpr> si la valeur logique de <lexpr> est vraie (TRUE) et <fexpr> dans tous les autres cas.

Erreurs de la Boîte de calcul

Le tableau 20-2 répertorie les erreurs détectées par la Boîte de calcul.

Tableau 20-2. Erreurs de la Boîte de calcul

Message d'erreur	Signification du message d'erreur
erreur de syntaxe	Opérateur mal utilisé, etc.
symbole incorrect	Caractère non reconnu.
variable de sortie demandée	Impossible d'affecter à une variable d'entrée.
il manque une variable de sortie	Tentative d'affectation à une variable de sortie inexistante.
il manque une variable	Référence à une variable d'entrée ou de sortie inexistante.
pas assez d'arguments	Arguments insuffisants pour une fonction.
trop d'arguments	Trop d'arguments pour une fonction.
liste d'arguments inachevée	La formule s'est terminée avant la parenthèse de fin de la liste des arguments.
il manque une parenthèse à gauche	Le nom de la fonction n'est pas suivi par une liste d'arguments.
il manque une parenthèse à droite	La formule s'est terminée avant toutes les parenthèses de fin correspondantes.
il manque deux-points	Utilisation incorrecte de l'opérateur ternaire conditionnel.

Tableau 20-2. Erreurs de la Boîte de calcul (Suite)

Message d'erreur	Signification du message d'erreur
il manque un point-virgule	L'instruction de la formule n'est pas terminée par un point-virgule.
il manque le signe égal	L'instruction de la formule ne constitue pas une affectation correcte.

VI Serveur

Ce chapitre présente le mécanisme de contrôle des VIs et des applications. Vous y trouverez également une présentation des procédures de contrôle à distance des VIs ou applications.

Le G propose un accès par programme à une grande partie de ses fonctions par le biais du VI Serveur. Vous pouvez accéder à ces fonctions à partir de n'importe quel client ActiveX sur les plates-formes Windows 95/NT. Vous pouvez également accéder à toutes les possibilités du VI Serveur, sur toutes les plates-formes, par le biais de certaines fonctions du langage G. Ces fonctions permettent également d'exécuter la majorité des opérations du VI Serveur dans une version locale de BridgeVIEW ou de LabVIEW ainsi que n'importe quelle version à distance au travers d'un réseau TCP/IP. Pour plus d'informations sur les possibilités client et serveur ActiveX, consultez le *Manuel de l'utilisateur LabVIEW* ou le *BridgeVIEW User Manual* ainsi que la *Référence en ligne* de votre logiciel.

Utilisation du VI Serveur

Voici quelques-unes des tâches que vous pouvez effectuer en utilisant le VI Serveur :

- Vous pouvez charger des VIs en mémoire de manière dynamique plutôt que de les lier de manière statique dans votre application. Vous pourrez ensuite les appeler comme un sous-VI normal en utilisant les fonctions du VI serveur. Cela peut se révéler utile lorsque vous avez une application de taille importante et que vous souhaitez libérer de la mémoire ou réduire le temps de démarrage. En rendant certaines parties de votre application rarement utilisées (comme un ensemble de boîtes de dialogue de configuration) chargeables et exécutables sur demande, vous pouvez réduire l'utilisation de la mémoire par votre application ainsi que le temps nécessaire à son chargement en mémoire. Lorsque l'utilisateur a terminé les opérations, vous pouvez libérer les VIs et augmenter la mémoire disponible.
- Vous pouvez contrôler par programme l'apparence de l'interface utilisateur d'un VI. Par exemple, vous pouvez déterminer de manière dynamique l'emplacement d'une fenêtre de VI ou faire défiler un

panneau pour qu'une zone reste visible en permanence ou fermer ou ouvrir sa fenêtre. Toutes ces propriétés de la face-avant du VI peuvent être contrôlées par le biais du VI Serveur.

- Vous pouvez facilement créer une application serveur qui exporte des fonctions pouvant être appelées à partir de BridgeVIEW ou de LabVIEW sur Internet. Par exemple, vous pouvez avoir une application d'acquisition des données qui acquiert et enregistre des données sur un site à distance. Vous pouvez ainsi échantillonner ces données de temps en temps à partir de votre machine locale. Par un simple paramétrage des préférences, vous pouvez rendre certains VIs accessibles sur Internet afin que le transfert des données les plus récentes soit aussi facile que l'appel d'un sous-VI. Le VI Serveur s'occupe de tous les détails relatifs au réseau et réalise l'opération quelle que soit la plate-forme du client ou du serveur.
- Vous pouvez modifier les propriétés d'un VI par programme et enregistrer ces modifications. Par exemple, pendant le développement de votre application, vous voudrez peut-être que les VIs soient configurés de manière à ce que la mise au point et les menus locaux exécutables soient disponibles, les barres de défilement visibles et les fenêtres redimensionnables. Toutefois, lorsque vous distribuez votre application, vous devez désactiver ces fonctions et vous assurer que certaines des autres propriétés sont correctement définies. Par exemple, vous devez vérifier si un VI est ré-entrant, le système d'exécution sous lequel le VI peut être exécuté et le chemin vers le fichier d'aide du VI. Toutes ces propriétés peuvent être définies par programme et interrogées par le VI Serveur, vous permettant ainsi d'écrire des applications qui éditent les VIs plutôt que d'utiliser la boîte de dialogue "Configuration du VI" pour chacun d'entre eux.
- Vous pouvez créer une architecture modulaire pour votre application afin de pouvoir rajouter une fonctionnalité après sa distribution aux clients. Par exemple, vous avez peut-être un jeu de VIs de filtrage de données ayant tous les mêmes paramètres. En concevant votre application de manière à charger ces filtres de manière dynamique à partir d'un répertoire de modules, vous pouvez envoyer votre application avec une partie de ces filtres et proposer des options de filtrage supplémentaires aux utilisateurs en plaçant simplement les nouveaux VIs de filtrage dans le répertoire de modules.

Fonctions du VI Serveur

Vous pouvez accéder aux fonctions VI Serveur par le biais de références à deux classes d'objets principales : l'objet Application et l'objet VI. Une fois que vous avez créé une référence à l'un de ces objets, vous pouvez le transmettre à une fonction à exécuter sur l'objet et, lorsque vous en avez terminé avec cet objet, vous pouvez refermer la référence. Cette convention de programmation est identique aux références d'E/S sur fichiers et aux références de réseau.

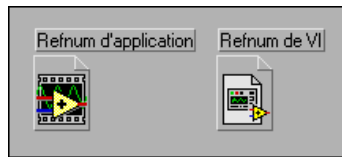
La transparence du réseau est une caractéristique importante des références d'application et de VI. Cela signifie que vous pouvez ouvrir des références d'objets sur des machines à distance de la même manière que les références d'objets sur votre propre machine. Une fois que vous avez ouvert une référence d'un objet à distance, vous pouvez le traiter exactement comme un objet local, à quelques exceptions près. Pour des opérations sur un objet à distance, le VI Serveur s'occupe de l'envoi des informations concernant l'opération au travers du réseau et de la récupération des résultats. Votre programme ne change pas et ne fait aucune différence entre l'opération à distance ou locale.

Avec une référence d'une application G, vous pouvez obtenir les informations concernant l'environnement du G telles que la nature de la plate-forme sur laquelle votre application est exécutée, la version de BridgeVIEW ou de LabVIEW ou la liste de tous les VIs actuellement en mémoire. Vous pouvez également définir des informations telles que le nom de l'utilisateur en cours ou la liste des VIs exportés vers d'autres applications.

Lorsque vous définissez une référence de VI, vous le chargez en mémoire. Une fois que la référence est établie, le VI reste en mémoire jusqu'à ce que vous fermiez la référence. Plusieurs références d'un VI peuvent être ouvertes simultanément. Dans ce cas, le VI reste en mémoire jusqu'à ce que toutes les références du VI soient fermées. Avec une référence d'un VI, vous pouvez obtenir et définir toutes les propriétés du VI disponibles dans la **Configuration du VI**, ainsi que les propriétés dynamiques telles que la position de la fenêtre de la face-avant. Vous pouvez également imprimer le VI par programme, l'enregistrer vers un autre emplacement et exporter et importer ses chaînes de caractères afin de les convertir.

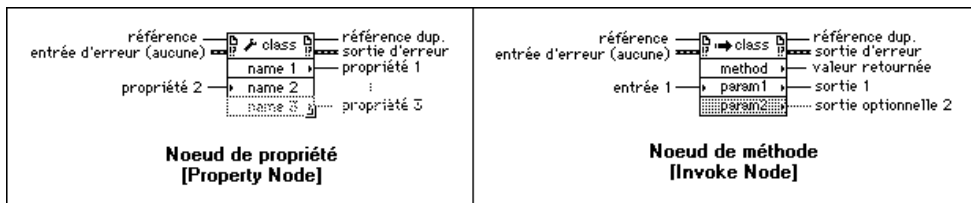
Références d'application et de VI

Dans la plupart des applications, vous pouvez créer une référence d'un VI en utilisant respectivement la fonction "Ouvrir une référence d'application" et "Ouvrir une référence de VI". Si vous souhaitez afficher la référence sur la face-avant, sélectionnez **Chemin & Refnum** dans la palette **Commandes**. Choisissez le refnum de VI ou d'application. Le type de données de ce refnum s'affiche par défaut comme un refnum d'application. Si vous voulez le modifier en refnum du VI, ouvrez le menu local du refnum et choisissez **Sélectionner une classe de VI Serveur» Virtual Instrument**.



Utilisation des nœuds de propriété et de méthode avec les références d'application et de VI

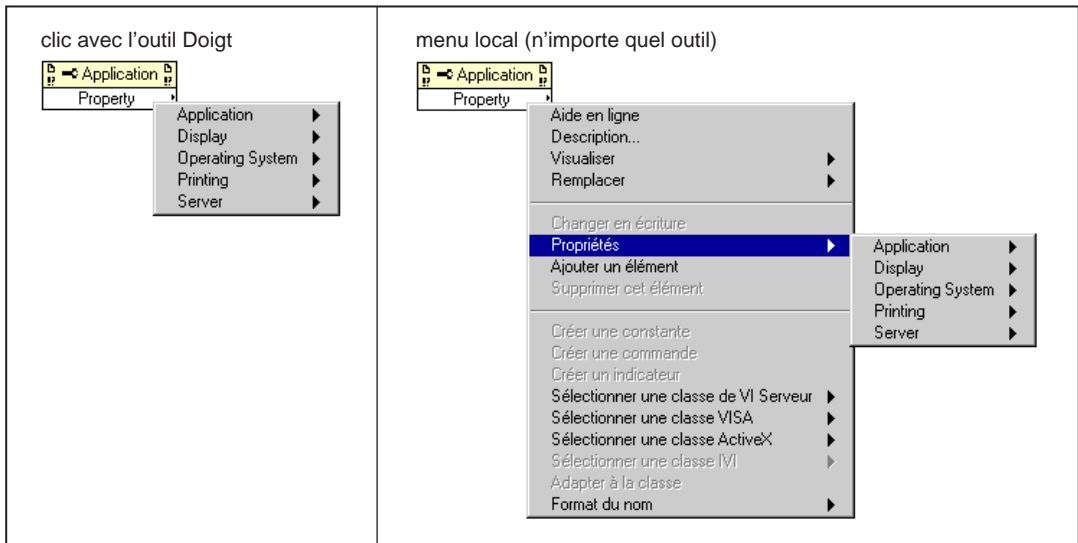
De nombreuses opérations parmi celles disponibles sur les références d'application et de VI ne le sont que par le biais des nœuds de propriété et de méthode. Ces nœuds sont identiques et ont deux entrées et deux sorties dans la partie supérieure du nœud et une liste de variables d'entrées et de sorties dans la partie inférieure.



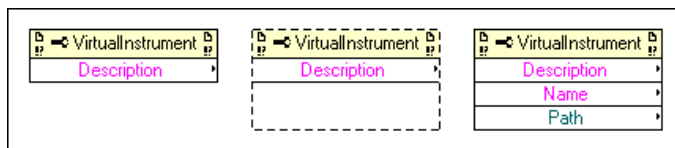
Référence est une référence d'application ou de VI et **référence dup** est une copie de la référence utilisée pour transmettre la valeur à d'autres fonctions. **Entrée d'erreur** est le cluster d'erreur standard qui contient les informations sur l'erreur, le code de l'erreur et une description de l'erreur. Si une erreur est présente dans l'**entrée d'erreur**, le nœud ne s'exécute pas et passe simplement l'erreur à la **sortie d'erreur**. Si une erreur se produit pendant l'opération du nœud, la **sortie d'erreur** contient les informations relatives à l'erreur.

Les deux nœuds sont polymorphes en ce qui concerne l'entrée de référence. Lorsque vous reliez un refnum d'application ou de VI à cette entrée, le nœud s'adapte automatiquement au type de données et ne rend disponibles que les opérations applicables à ce type.

Avec le nœud de Propriété, vous pouvez obtenir (lire) et définir (écrire) les différentes propriétés d'une application ou d'un VI. Vous pouvez sélectionner les propriétés dans le menu local du nœud en cliquant simplement sur un terminal de propriété avec l'outil Doigt ou en ouvrant un menu local sur le terminal.

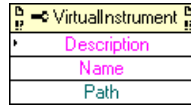


Vous pouvez obtenir ou définir plusieurs propriétés en n'utilisant qu'un seul nœud. Au fur et à mesure que vous agrandissez le nœud de propriété, de nouveaux terminaux sont ajoutés.



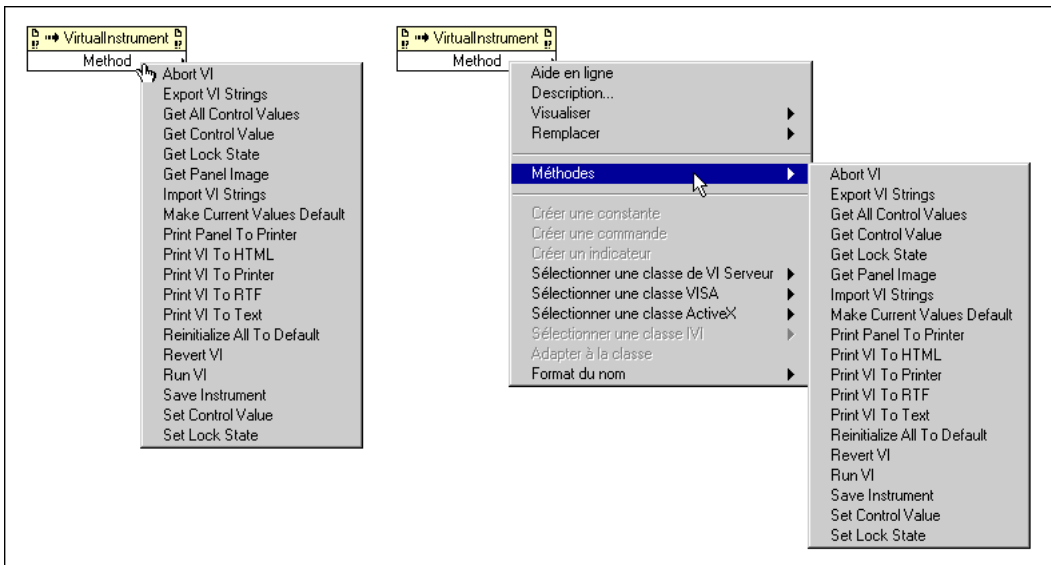
Dans un nœud de propriété, vous pouvez lire et écrire des propriétés, comme le montre l'illustration suivante. Les propriétés que vous lisez sont indiquées par une petite flèche à droite, tandis que les propriétés que vous écrivez ont une petite flèche à gauche. Vous pouvez choisir de lire ou

d'écrire des propriétés en sélectionnant **Changer en lecture** ou **Changer en écriture** dans le menu local.

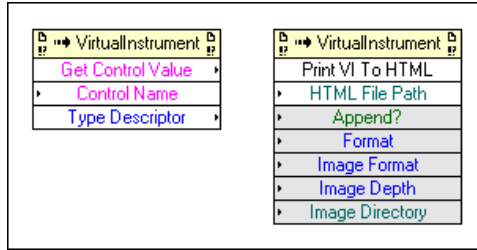


Le nœud s'exécute de haut en bas. Si une erreur se produit au milieu du nœud, les propriétés restantes sont ignorées et une erreur est retournée. La chaîne de caractères de l'erreur contient les informations concernant la propriété qui a provoqué l'échec.

Avec le nœud de méthode, vous pouvez effectuer des actions ou des méthodes sur une application ou un VI. Contrairement au nœud de propriété, un seul nœud de méthode n'exécute qu'une seule méthode sur une application ou un VI. Vous pouvez accéder aux méthodes disponibles en cliquant sur le nœud de méthode avec l'outil Doigt ou en ouvrant un menu local sur le terminal, comme pour le nœud de propriété.



Le nom de la méthode constitue toujours le premier terminal de la liste des paramètres. Si la fonction retourne une valeur, ce résultat est la valeur du terminal de la méthode. Par exemple, une valeur est retournée par la méthode **Obtenir la valeur de la commande**. Sinon, le terminal de la méthode n'a pas de valeur, comme pour **Imprimer le VI en HTML**.



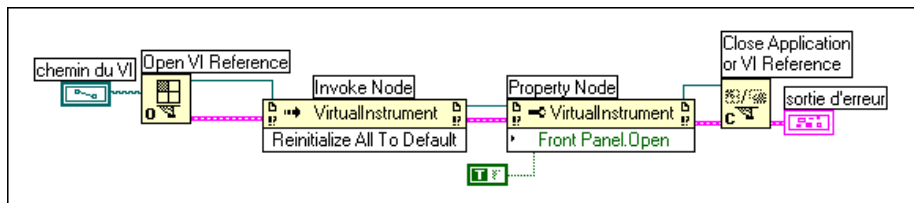
Le nœud de méthode liste les paramètres de haut en bas avec les paramètres optionnels grisés en bas. Dans la précédente illustration, **Chemin du fichier HTML** est un paramètre nécessaire, alors que **Ajouter?**, **Format**, **Format de l'image**, **Profondeur de l'image**, et **Répertoire de l'image** sont des paramètres optionnels.

Exemple de propriétés et méthodes de classe de VI et d'application

Les sections suivantes traitent des applications des classes d'application et de VI communes. Pour observer d'autres applications où les méthodes et les propriétés de classe d'application et de VI sont utilisées, consultez la bibliothèque `examples\general\viserver`.

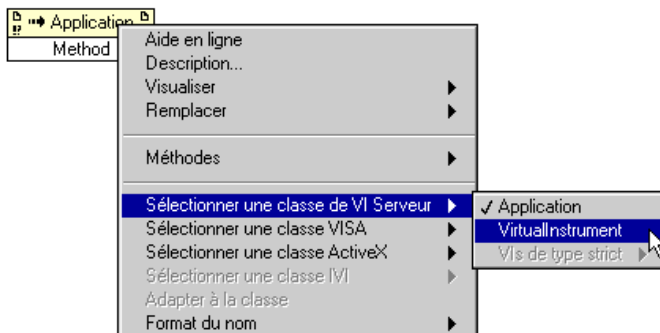
Manipulation des méthodes et propriétés de classe de VI

Vous pouvez définir ou obtenir des propriétés sur un VI indépendamment de l'exécution des méthodes. Dans certaines applications, vous devez effectuer les deux, accéder aux propriétés du VI et exécuter une méthode sur un VI. Dans le diagramme suivant, les objets de la face-avant d'un VI sont réinitialisés à leurs valeurs par défaut, puis la face-avant est ouverte pour afficher ces valeurs.



Avant d'accéder aux propriétés et aux méthodes d'un VI, vous devez créer une référence de ce VI en exécutant "Ouvrir une référence de VI". Pour appeler une méthode d'un VI, utilisez le nœud de méthode. Une fois la connexion établie entre "Ouvrir une référence de VI" et le nœud de méthode, vous pouvez accéder à toutes les méthodes de classe du VI.

Vous pouvez également ouvrir le menu local du nœud et choisir **Sélectionner une classe de VI Serveur»Virtual Instrument**.

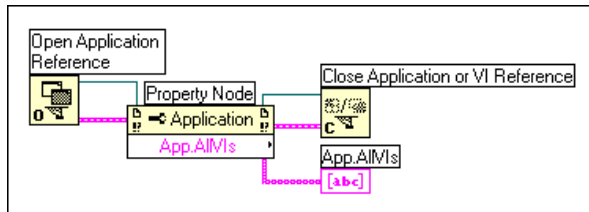


Le nœud de propriété opère de manière identique au nœud de méthode. Une fois que vous reliez une référence de VI au nœud de propriété, vous pouvez accéder à toutes les propriétés de classe de VI. De même, vous pouvez ouvrir le menu local du nœud et choisir **Sélectionner une classe de VI Serveur»Virtual Instrument**.

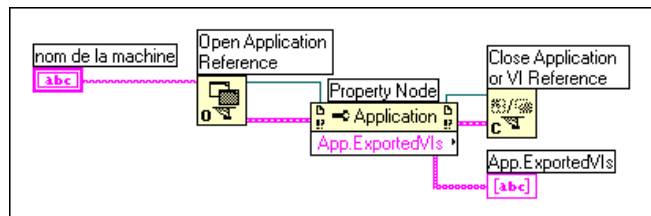
Pour définir les valeurs de propriété, assurez-vous que la flèche se trouve à gauche en ouvrant un menu local et en sélectionnant "Changer en écriture". Vérifiez toujours la possibilité d'erreurs. Les nœuds de propriété et de méthode ne s'exécutent pas si une erreur se produit avant l'exécution. Si une erreur se produit dans un de ces nœuds, la propriété ou la méthode où l'erreur se produit apparaît dans le message d'erreur.

Manipulation des méthodes et des propriétés de classe d'application

Vous pouvez définir ou obtenir des propriétés sur un LabVIEW local ou à distance indépendamment de l'exécution des méthodes de LabVIEW. Dans le diagramme suivant, les VIs en mémoire sur une machine locale sont affichés sur la face-avant dans un tableau de chaînes de caractères.



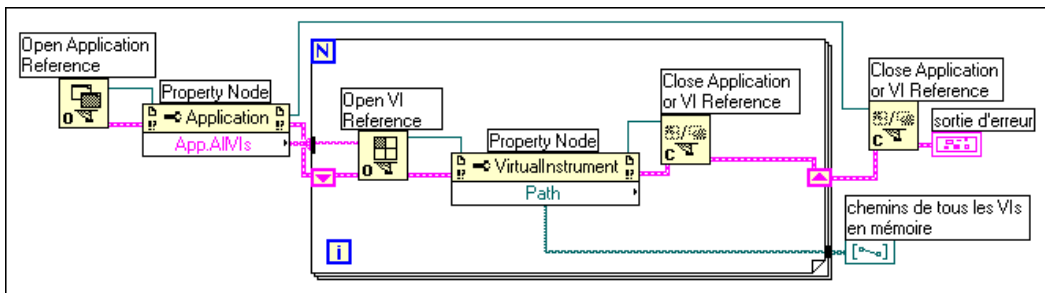
Si vous voulez rechercher des VIs en mémoire sur une machine à distance, reliez une commande chaîne de caractères à l'entrée du nom de la machine et entrez l'adresse IP numérique de la machine ou le nom de domaine, comme le montre l'illustration suivante. Vous devez également modifier la propriété en **VIs exportés en mémoire** puisque la propriété **Tous les VIs en mémoire** utilisée dans la précédente illustration ne s'applique qu'aux versions locales de LabVIEW et de BridgeVIEW.



Vérifiez toujours la possibilité d'erreurs. Le nœud de propriété ne s'exécute pas si une erreur se produit avant l'exécution. Si une erreur se produit, la propriété où l'erreur se produit apparaît dans le message d'erreur.

Manipulation des méthodes et des propriétés de classe d'application et de VI

Vous pouvez utiliser les méthodes et les propriétés de classe d'application et de VI de manière séparée. Dans certaines applications, vous devez accéder aux propriétés et aux méthodes pour les deux classes. Dans le diagramme suivant, les VIs en mémoire sur une machine locale sont déterminés et le chemin vers chacun de ces VIs est affiché 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 chaque VI, vous devez accéder à une propriété de classe du VI. Le nombre de VIs en mémoire détermine le nombre de fois que la boucle For s'exécute. "Ouvrir une référence de VI" et "Fermer une référence de VI" doivent se trouver à l'intérieur de la boucle For car il vous faut une référence de VI pour chaque VI en mémoire. Il est préférable de ne pas fermer la référence d'application tant que tous les chemins de VI ne sont pas rassemblés.



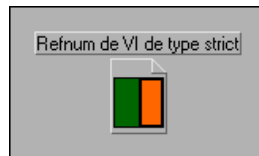
Refnums de VI de type strict

Les refnums de type strict ne doivent être utilisés que dans des applications où vous effectuez un appel dynamique à un VI. Dans ces applications, il existe deux situations différentes où ils sont utilisés. La première, qui est probablement la plus courante, est lorsque vous souhaitez transmettre une référence de VI de type strict dans un sous-VI sous forme de paramètre. Dans ce cas, vous devez connecter la commande de refnum de type strict au cadre connecteur du VI et relier le terminal de refnum à l'entrée d'un nœud d'Appel par référence. La valeur du refnum est utilisée dans ce cas afin de déterminer quel VI est appelé.

La deuxième situation se présente lorsque vous reliez un refnum de VI de type strict à l'entrée de spécification de type de la fonction "Ouvrir une référence de VI". Dans ce cas, la valeur de la commande est ignorée : seul

le type du refnum est utilisé par cette fonction. Il sert à déterminer si le VI qui est ouvert possède le même cadre connecteur que le refnum de VI de type strict.

Pour créer un refnum de type strict, déposez un refnum de VI sur la face-avant en sélectionnant **Commandes»Chemin & Refnum»Refnum de VI ou d’application**. Ouvrez le menu local et choisissez **Sélectionner une classe de VI Serveur»Parcourir...**. La boîte de dialogue **Ouvrir un VI** s’affiche et vous invite à choisir un VI.

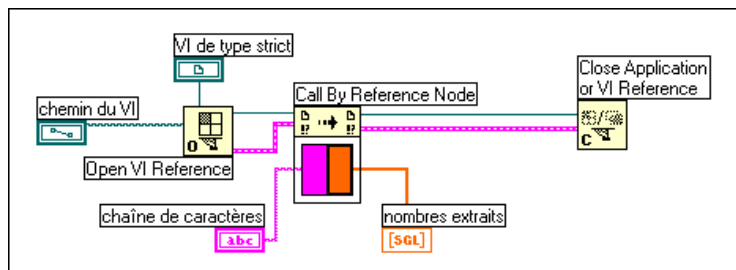


Souvenez-vous que même si vous spécifiez un VI pour des refnums de type strict, il ne stocke que les informations du cadre connecteur. En effet, aucune association permanente n’est créée entre le refnum et le VI. De plus, ne confondez pas la sélection du cadre connecteur du VI avec l’obtention d’une référence du VI sélectionné. Vous devez spécifier un VI particulier par le biais de l’entrée du **chemin du VI** sur la fonction “Ouvrir une référence de VI”.

Une fois que vous avez sélectionné les cadres connecteurs des refnums de type strict, le cadre connecteur est retenu dans le sous-menu du refnum de **VI Sélectionner une classe de VI Serveur»VIs de type strict**. Si vous quittez BridgeVIEW ou LabVIEW, ces sélections de connecteur ne seront plus retenues au prochain démarrage de l’application.

Exemple de refnums de VI de type strict

La seule application où vous devez utiliser des refnums de VI de type strict consiste à appeler un VI de manière dynamique. Le diagramme suivant vous montre un exemple.



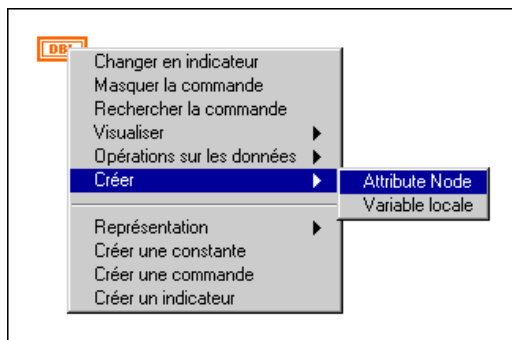
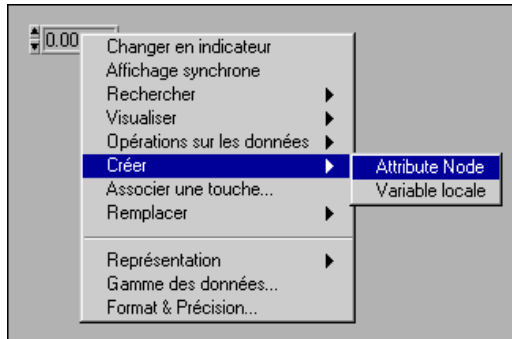
Une fois que vous reliez le **chemin du VI** et **VI de type strict** à la fonction “Ouvrir une référence de VI”, vous pouvez connecter cette dernière au nœud d’appel par référence. Le cadre connecteur s’affiche automatiquement dans le nœud d’appel par référence. Vous pouvez ensuite relier n’importe quelles valeurs d’entrée et d’affichage de sortie au cadre connecteur. Le nœud d’appel par référence appelle le VI spécifié par l’entrée de référence du VI en utilisant n’importe quelles valeurs d’entrée spécifiées. Ce nœud est utile lorsque vous ne souhaitez pas charger tous les sous-VIs en mémoire en une seule fois.

Attribute Nodes

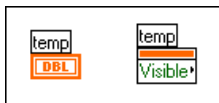
Ce chapitre présente l'utilisation des attribute nodes afin de définir et de lire par programme des attributs de commandes de face-avant. Parmi ces attributs utiles, vous trouverez les couleurs d'affichage, la visibilité de la commande, les chaînes de caractères de menu pour une commande de type menu déroulant, les couleurs des tracés des graphes et des graphes déroulants, et les curseurs de graphe.

Création d'attribute nodes

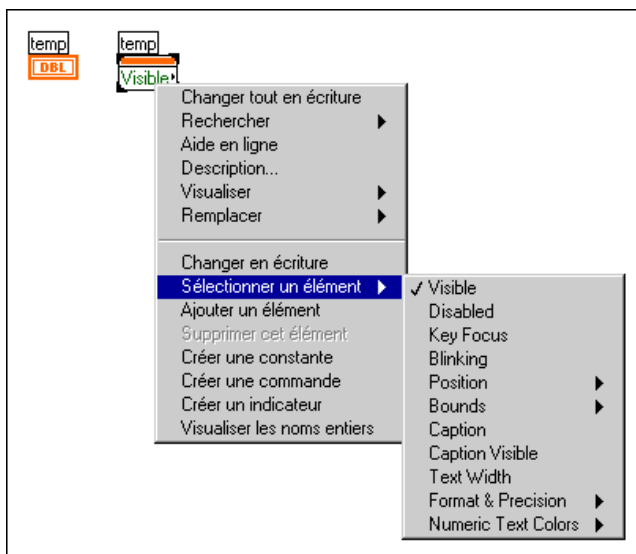
Vous pouvez créer un attribute node en sélectionnant l'élément **Créer» Attribute Node** dans le menu local d'une commande de face-avant ou d'un terminal de commande.



En sélectionnant cet élément, vous créez un nouveau nœud sur le diagramme situé à côté du terminal de la commande, comme le montre l'illustration suivante. Si la commande possède une étiquette qui lui est associée, l'étiquette de la commande est utilisée pour l'étiquette initiale de l'attribute node. Vous pouvez modifier l'étiquette après la création du nœud.

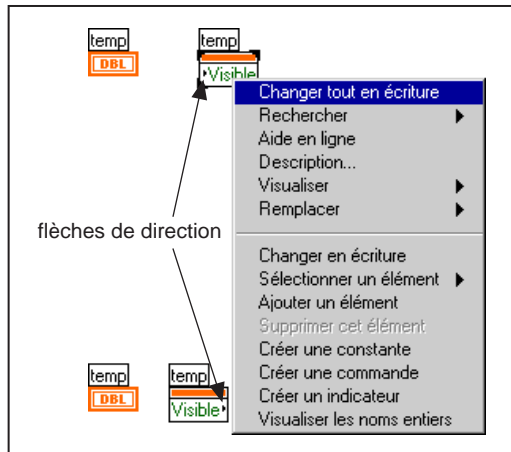


Si vous ouvrez un menu local du terminal d'attribut et choisissez ensuite **Sélectionner un élément**, vous verrez un menu d'attributs que vous pouvez écrire ou lire sur la commande, comme le montre l'illustration suivante. Vous pouvez utiliser un raccourci vers la liste des attributs en cliquant sur l'attribute node avec l'outil Doigt.

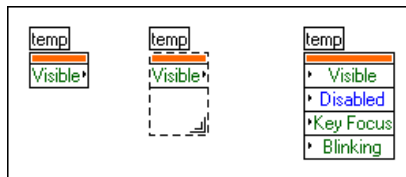


Vous pouvez choisir de lire ou d'écrire des attributs en sélectionnant l'élément **Changer en lecture** ou l'élément **Changer en écriture** dans le menu local de l'attribute node, comme indiqué dans l'illustration suivante.

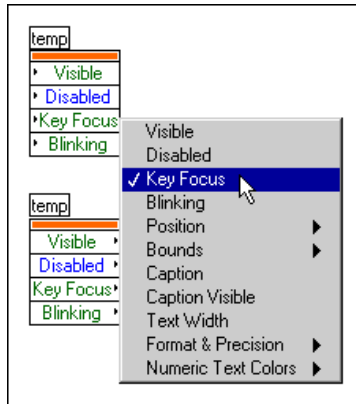
Vous pouvez écrire un attribut lorsque la petite flèche se situe à gauche du terminal. Vous pouvez lire un attribut lorsque la flèche se situe à droite du terminal.



Vous pouvez lire ou écrire plusieurs attributs avec le même nœud en agrandissant l'attribut node. De nouveaux terminaux sont ajoutés au fur et à mesure que le nœud s'agrandit. L'ordre d'exécution d'un attribut node est du haut vers le bas.

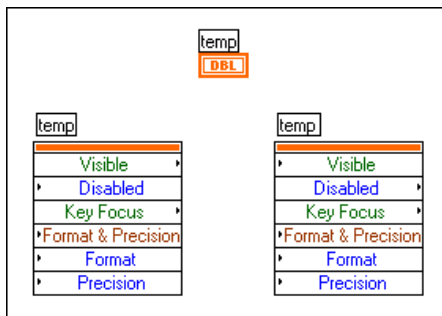


Vous pouvez associer un terminal à un attribut donné en cliquant sur le terminal avec l’outil Doigt et en sélectionnant un attribut dans le menu local de l’attribut node.

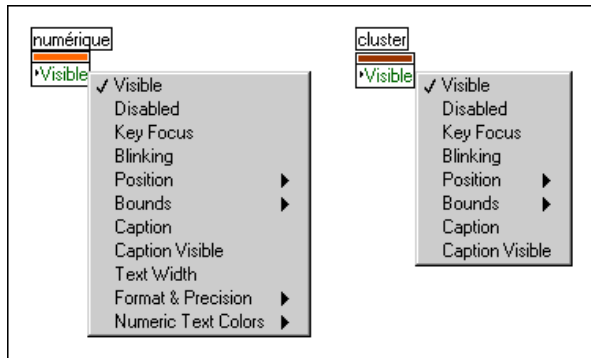


Vous pouvez créer plusieurs attribut nodes en clonant un nœud existant ou en sélectionnant à nouveau l’élément **Créer»Attribut Node**. Pour cloner un nœud existant, cliquez sur ce nœud et faites le glisser avec l’outil Flèche tout en maintenant la touche <Ctrl> (**Windows**), <option> (**Macintosh**), <meta> (**Sun**) ou <Alt> (**HP-UX**) enfoncée. Toutefois, si vous copiez et collez un attribut node en utilisant les commandes du menu **Edition**, une nouvelle copie de l’attribut node et de la commande à laquelle il se réfère est réalisée.

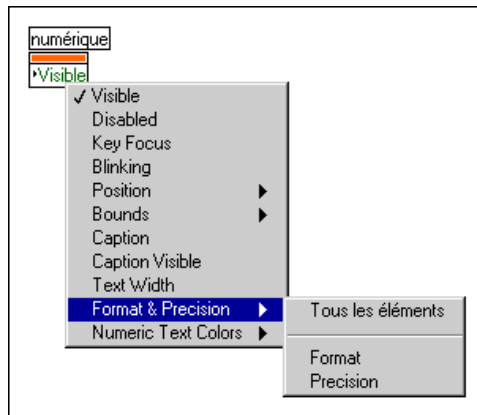
Chaque copie d’un attribut node peut indépendamment avoir des terminaux d’attribut pour la lecture et pour l’écriture.



L'illustration suivante montre les attributs d'un cluster et d'une commande numérique à l'intérieur du cluster.



Certaines commandes, telles que le graphe, possèdent un grand nombre d'attributs que vous pouvez soit lire soit définir. Certains de ces attributs sont groupés en catégories et répertoriés dans des sous-menus, telles que la catégorie **Format & Précision** pour une commande numérique. Vous pouvez choisir de définir tous les attributs en une seule fois en sélectionnant l'élément **Tous les éléments** dans le sous-menu. Vous pouvez également définir un ou plusieurs de ces éléments à la fois en sélectionnant les attributs spécifiques. L'illustration suivante montre l'exemple de l'élément **Format & Précision** d'une commande numérique.



Une fois que vous avez créé un attribut node, les éléments **Trouver la commande** et **Trouver le terminal** des menu locaux du terminal et de la commande sont transformés en sous-menus qui vous servent à rechercher des attribut nodes. De même, l'attribut node possède des options de recherche de commande et de terminal qui lui sont associées.

Pour obtenir une exemple d'utilisation d'attribut node, consultez la bibliothèque `examples\general\attribute.llb`.

Utilisation de l'aide d'attribut node

La fenêtre d'aide et la *Référence en ligne* (menu **Aide**) sont des outils inestimables pour l'utilisation des attribut nodes. Vous pouvez les utiliser pour rechercher des descriptions, des types de données et des valeurs valides pour les attributs. Pour plus d'informations, consultez la section *Aide pour les attributs* du chapitre 1, *Introduction à la programmation en G*.

Attributs de base

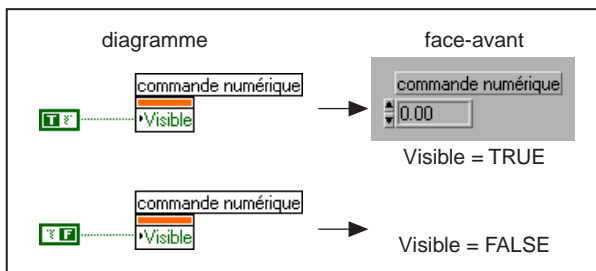
De nombreux attributs sont disponibles pour la modification des différents objets de face-avant de votre application. Cette section traite des attributs Visible, Désactiver, Focus de la touche, Clignotement, Position et Limites qui sont identiques à presque tous les objets de face-avant.

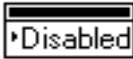


Attribut Visible

L'attribut Visible vous permet de lire ou de définir la visibilité d'un objet de face-avant. L'objet associé est visible lorsque l'attribut est vrai (TRUE) et masqué lorsque l'attribut est Faux (FALSE).

Dans l'illustration suivante, la commande numérique est définie en un état invisible. Une valeur booléenne Vrai (TRUE) rend la commande à nouveau visible, de la manière suivante.

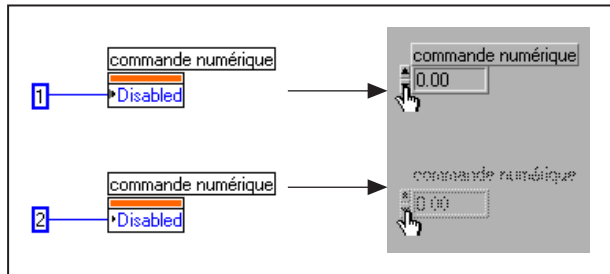




Attribut Désactiver

L'attribut Désactiver vous permet de contrôler l'accès à un objet que peut avoir un utilisateur. Une valeur de zéro active un objet de manière à ce que l'utilisateur puisse l'exécuter. Une valeur de 1 désactive l'objet, ce qui empêche l'opération. Une valeur de 2 désactive et grise l'objet.

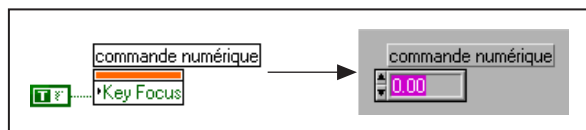
Vous pouvez désactiver l'accès de l'utilisateur à une commande numérique. La commande ne change pas d'apparence lorsqu'elle est désactivée dans le premier exemple. Dans le deuxième exemple, l'accès de l'utilisateur à la commande numérique est désactivé et la commande est grisée.



Attribut “Focus de touche”

L'attribut “Focus de touche” vous permet de transformer une commande en focus de touche ou de vérifier si elle possède actuellement le focus. Une commande à focus de touche se comporte comme si vous l'aviez sélectionnée en appuyant sur la touche <Tab>. Avec la plupart des commandes, vous pouvez entrer des valeurs en les tapant à l'aide du clavier. Vous pouvez également définir le focus de touche sur une face-avant en appuyant sur la touche <Tab> en mode d'exécution ou en appuyant sur la touche d'accès rapide associée à la commande (affectée par l'utilisation de l'élément **Utilisation du clavier**).

Vous pouvez transformer une commande numérique en focus de touche. Vous pouvez ensuite entrer une nouvelle valeur dans la commande sans devoir la sélectionner avec la souris.

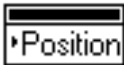
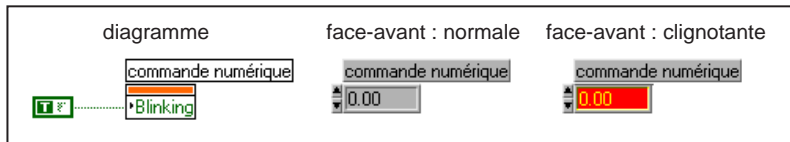




Attribut Clignotement

L'attribut Clignotement vous permet de lire ou de définir l'état de clignotement d'un objet. Si vous définissez cet attribut en vrai (TRUE), un objet de face-avant clignote. La vitesse de clignotement et les couleurs sont définies dans la boîte de dialogue **Préférences**. Lorsque vous définissez cet attribut en faux (FALSE), l'objet arrête de clignoter.

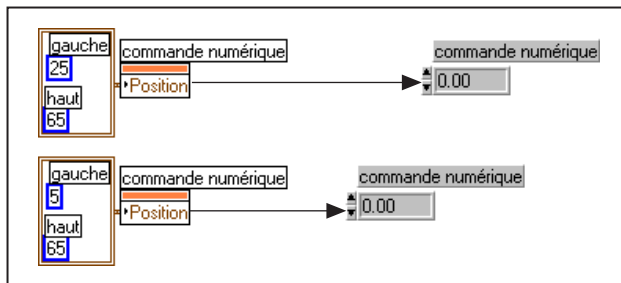
Dans l'illustration suivante, l'indicateur de face-avant est défini de manière à clignoter.

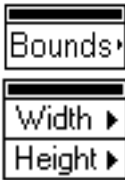


Attribut Position

L'attribut Position vous permet de définir ou de lire la position du coin supérieur gauche d'un objet sur la face-avant. La position est déterminée en unités de pixels par rapport à l'origine de la face-avant, qui se situe à l'extrémité supérieure gauche de la fenêtre. Elle peut se trouver ailleurs si vous avez fait défiler la fenêtre. Cet attribut comporte un cluster de deux entiers longs non signés. Le premier élément du cluster (gauche) est l'emplacement du côté gauche de la commande et le deuxième élément du cluster (haut) est l'emplacement du côté supérieur de la commande par rapport à l'origine de la face-avant.

En exécutant ou en utilisant l'attribut node Position, la commande numérique change d'emplacement sur la face-avant dans l'illustration suivante.



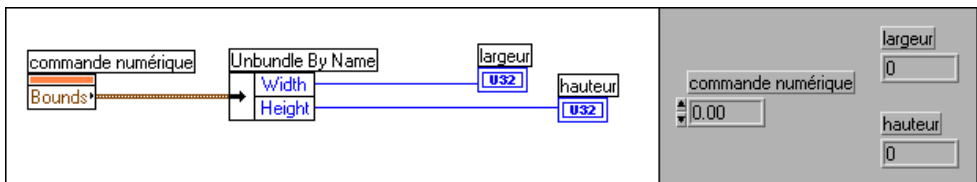


Attribut Limites (lecture seule)

L'attribut Limites lit la limite d'un objet sur la face-avant en unités de pixels. La valeur représente l'objet et toutes ses parties, y compris l'étiquette, la légende, l'échelle, etc. Cet attribut comporte un cluster de deux entiers longs non signés. Le premier élément du cluster (largeur) est la largeur de l'objet en pixels et le deuxième élément du cluster (hauteur) est la hauteur de l'objet en pixels.

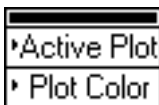
L'attribut Limites est un attribut en *lecture seule*. Il *ne* redimensionne *pas* une commande ou un indicateur sur la face-avant. La plupart des objets possèdent d'autres attributs pour le redimensionnement, tels que l'attribut "Plot Area Size" pour les graphes et les graphes déroulants. Cet attribut est utile lorsque vous devez lire l'intégralité de la taille d'un objet, y compris toutes les parties optionnelles. Vous pouvez ainsi positionner d'autres objets en rapport avec l'objet sur lequel vous travaillez.

Vous pouvez déterminer les limites d'une commande numérique, comme le montre l'illustration suivante.



Exemples d'attributs spécifiques à certaines commandes ou indicateurs

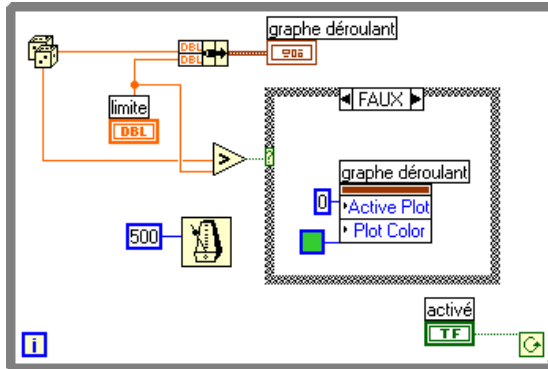
Les sections suivantes présentent des applications communes des attributs nodes. Consultez le répertoire `general` pour obtenir des exemples d'autres utilisations de l'attribut node.



Modification de la couleur du tracé sur un graphe déroulant

Les attributs illustrés à gauche définissent ou lisent le numéro de la courbe active (le tracé pour lequel les attributs ultérieurs spécifiques sont définis ou lus) et la couleur de la courbe pour la courbe active. La Courbe active est un entier qui indique quel tracé d'un graphe déroulant à plusieurs tracés vous voulez activer. La Couleur du tracé est un entier qui représente la couleur désirée. Vous pouvez accéder à la Couleur du tracé en sélectionnant **Plot Info»Plot Color** dans la liste des attributs.

Cet attribut node modifie la couleur du tracé spécifié par l'attribut Courbe active. Dans l'exemple suivant, la couleur du tracé à nombre aléatoire change de couleur lorsque les valeurs dépassent la limite définie par l'utilisateur. Remarquez que la courbe active doit être spécifiée avant que la couleur du tracé ne soit modifiée.



Définition des chaînes de caractères d'un attribut de booléen

Cet attribut de booléen définit ou lit les étiquettes d'une commande booléenne. L'entrée est un tableau capable de contenir quatre chaînes de caractères qui correspondent aux états False, True, pistage True et pistage False.

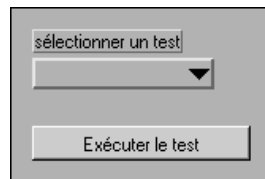
- True et False : états Marche et Arrêt du booléen.
- Passage True et passage False : niveaux de transition temporaire entre les états du booléen. Pistage True représente la transition lorsque le booléen passe de True à False. Le passage False représente le passage de False à True lorsque l'attribut booléen est modifié. Le repérage ne s'applique qu'aux booléens ayant l'action mécanique "Commutation au relâchement" et "Armement au relâchement". Ces actions mécaniques sont en état transitoire tant que vous n'avez pas relâché le bouton de la souris. Les chaînes de caractères de texte de passage True et False s'affichent pendant l'état transitoire.

Vous pouvez définir les chaînes de caractères affichées pour les commandes booléennes par le choix des chaînes de caractères **Stop**, **Exécuter**, et **Stop ?** et **Exécuter ?**.

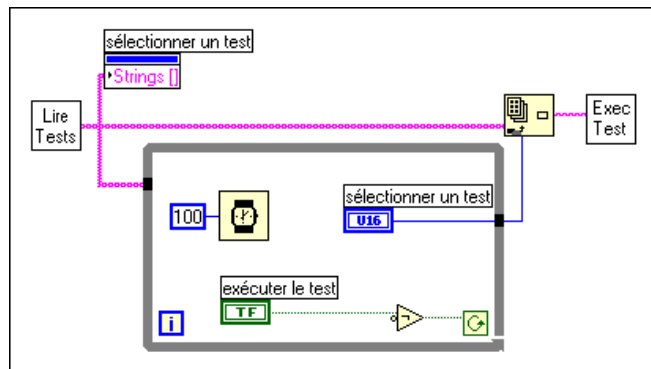
Définition des chaînes de caractères des commandes de type menu déroulant

La commande de type menu déroulant est une commande de menu local qui conserve la valeur numérique de l'élément actuellement sélectionné. Vous pouvez l'utiliser afin de présenter une liste d'options à l'utilisateur. Si les options ne peuvent être déterminées avant l'exécution, vous pouvez utiliser l'attribut node pour définir les options. Pour plus d'informations concernant les commandes de type menu déroulant, consultez le chapitre 13, *Commandes et indicateurs de type liste et menu déroulants*.

Dans l'exemple suivant, l'utilisateur voit un panneau avec une commande de type menu déroulant qui affiche une liste de tests. L'utilisateur sélectionne un test et clique sur le bouton **Exécuter le test** pour continuer.



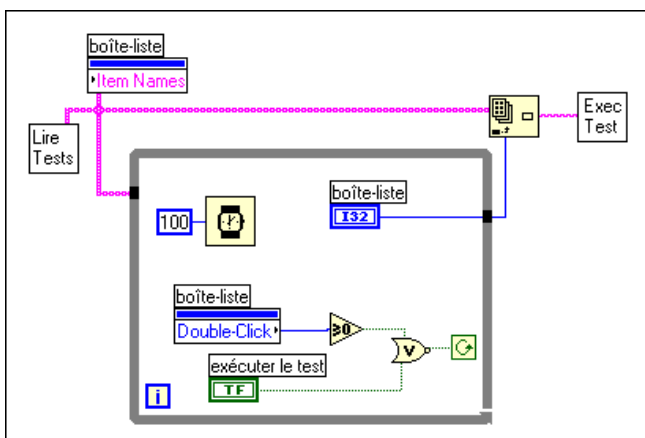
Le diagramme de l'illustration suivante lit une liste de tests valides dans un fichier et transmet la liste, représentée sous forme de tableau de chaînes de caractères, à un attribut node de la commande de type menu déroulant. Le diagramme effectue alors une boucle, en attendant que l'utilisateur clique sur le bouton **Exécuter le test**. Cette opération donne la possibilité à l'utilisateur de sélectionner un test dans la commande de type menu déroulant ou d'entrer des informations pour les autres commandes. Lorsque l'utilisateur sélectionne un test, la chaîne de caractères correspondant à la valeur numérique de la commande de type menu déroulant est lue et ensuite transmise à un VI qui exécute le test.



Utilisation d'un élément de liste déroulante à double-clic

L'attribut Double-clic est un attribut en lecture seule spécifique aux listes déroulantes. Cet attribut indique l'élément de la liste déroulante de la face-avant sur lequel vous devez double-cliquer. La valeur de l'attribut Double-clic est définie lorsque vous cliquez deux fois sur un élément ou appuyez sur <Entrée> (**Windows** ou **UNIX**) ou <Retour> (**Macintosh** et **Sun**) après avoir sélectionné un élément. La valeur de Double-clic est -1 si vous ne double-cliquez sur aucun élément. Elle revient à -1 une fois qu'elle a été lue. Elle revient également à -1 si vous sélectionnez un autre élément ou si vous définissez un des autres attributs.

L'illustration suivante propose un exemple d'utilisation de l'attribut node Double-clic afin de déterminer le test à exécuter.



La boucle s'arrête lorsque vous cliquez sur le bouton "Exécuter le test" ou lorsque vous double-cliquez sur un élément, auquel cas l'attribut Double-clic retourne le nombre de l'élément au lieu de -1.

Présentation d'options aux utilisateurs de manière sélective

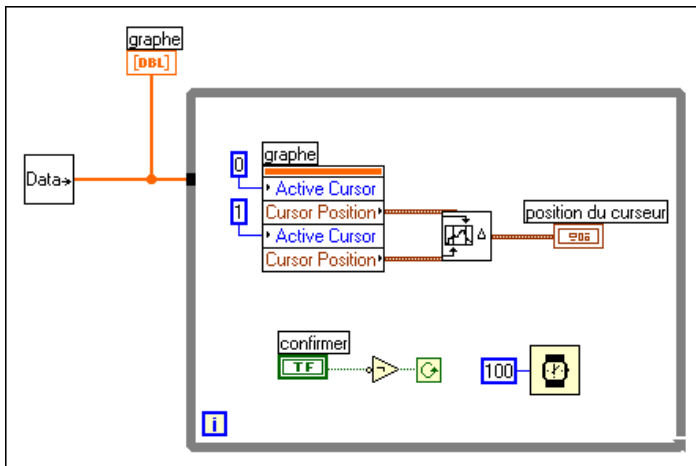
Lorsque les utilisateurs effectuent des sélections, vous voudrez peut-être leur présenter d'autres options. Vous avez trois possibilités.

1. Une des possibilités consiste à utiliser des sous-VIs locaux. Vous pouvez créer des sous-VIs avec les options que vous voulez présenter à l'utilisateur. En utilisant les éléments **Visualiser la face-avant lors de l'appel** et **Fermer après l'exécution si fermé à l'origine** de **Configuration du VI»Options d'exécution** au moment de la création de votre sous-VI, vous pouvez demander à ce que ces sous-VIs soient ouverts lors de l'appel.
2. Une autre méthode de présentation des options consiste à utiliser l'option `Visible` des `attribute nodes` afin d'afficher ou de masquer des commandes de manière sélective.
3. La troisième méthode consiste à utiliser l'option `Désactivé` des `attribute nodes` afin de désactiver des commandes de manière sélective. Lorsqu'une commande est désactivée, l'utilisateur ne peut pas en modifier la valeur.

Lecture par programme des curseurs du graphe

Vous pouvez utiliser des attributs afin d'accéder à des informations provenant d'un tracé sur un graphe multicourbes. Cependant, vous devez indiquer l'élément sur lequel opérer. Les curseurs multiples d'un graphe sont un bon exemple d'attribut qui doit être activé avant de devenir accessible à partir du diagramme.

Le diagramme suivant montre un VI qui affiche des données dans un graphe et lit la position des curseurs du graphe par programme.



En travaillant avec une boucle While, le VI active d'abord le curseur de la valeur min. et lit sa valeur numérique. Ensuite, le VI active et lit le curseur de la valeur max. Enfin, le VI calcule et affiche les informations concernant la sélection du curseur sur la face-avant. Lorsque vous appuyez sur le bouton Confirmer, le VI quitte la boucle et confirme les positions du curseur.

Observez les exemples dans la bibliothèque `exemples\general\graphs\zoom.llb` pour une application de lecture des curseurs d'un graphe par programme.

Variables globales et locales

Ce chapitre présente la définition et l'utilisation des variables locales et globales. Utilisez les variables globales pour accéder facilement à un jeu de valeurs à partir de plusieurs VIs. Les variables locales ont la même fonction, mais sont utilisées à l'intérieur d'un seul VI.

Les variables locales et globales sont des concepts avancés du G. Lisez attentivement tout ce chapitre avant de les utiliser.

Pour obtenir des exemples d'utilisation des variables locales et globales, consultez la bibliothèque `examples\general\globals.llb` et `examples\general\locals.llb`.

Variables globales

Une variable globale est un objet intégré au G. Lorsque vous créez une variable globale, un type spécial de VI est automatiquement créé. Vous ajoutez à ce VI des commandes de face-avant qui définissent les types de données des variables globales qu'il contient.

Il existe deux manières de créer des variables globales multiples. Vous pouvez soit créer plusieurs VIs, chacun ayant un élément, soit créer un VI à variables globales multiples en définissant plusieurs commandes sur la face-avant de la variable globale. La méthode du VI à variables globales multiples est plus efficace car elle vous permet de grouper les variables apparentées.

Vous pouvez créer une variable globale en la sélectionnant dans la palette **Fonctions»Structures** puis en la plaçant sur le diagramme.





Nœud de variable globale

Un nœud correspondant à la variable globale s’affiche sur le diagramme.

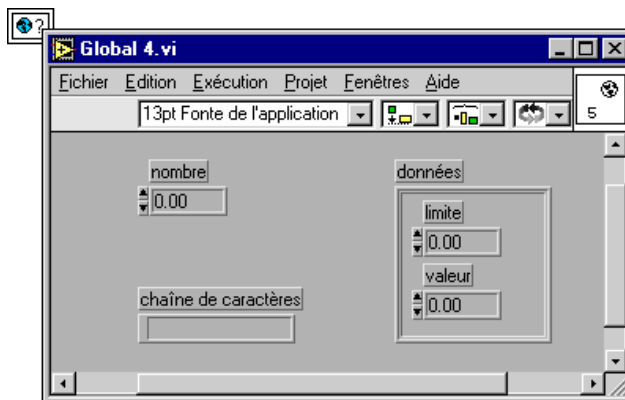
Double-cliquez sur le nœud pour ouvrir sa face-avant. Vous allez utiliser ce panneau pour définir les types de données d’une ou de plusieurs variables globales.



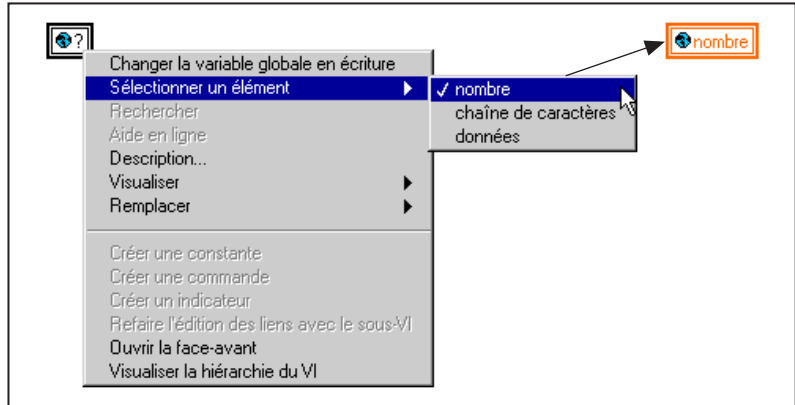
Remarque

Affectez un nom (étiquette) à chaque commande car vous devez faire référence à une variable globale spécifique en utilisant son nom. Une fois les types de données définis pour les variables globales, enregistrez le VI de variable globale.

L’illustration suivante est un exemple de face-avant qui décrit trois variables globales : un nombre, une chaîne de caractères et un cluster contenant deux valeurs.



Une fois que vous avez placé une variable globale sur un diagramme et que vous lui avez défini une face-avant, le nœud est associé à cette face-avant. Etant donné qu’un seul VI de variable globale peut définir plusieurs variables globales, vous devez sélectionner celle à laquelle vous voulez accéder à partir d’un nœud donné. Sélectionnez une variable globale en ouvrant le menu local du nœud et en sélectionnant l’élément par nom dans le menu **Sélectionner un élément** illustré ci-après. Ou cliquez sur le nœud avec l’outil Doigt et choisissez un élément.



Vous pouvez soit écrire dans une variable globale soit la lire. L'écriture dans une variable globale consiste à modifier sa valeur ; la lecture consiste à accéder à la variable globale comme à une source de données. Si vous voulez écrire dans une variable globale ou la lire, sélectionnez l'élément **Changer la variable globale en écriture** ou **Changer la variable globale en lecture** dans son menu local.

Les variables globales peuvent être écrites et lues par tout VI en mémoire. Il est important de connaître l'emplacement de tous les "lecteurs" et "éditeurs" de votre application pour qu'une variable globale ne soit pas modifiée à l'improviste. Les programmes en G peuvent gérer plusieurs activités à la fois et il n'est donc pas toujours facile de savoir quand l'accès à une variable globale par différents diagrammes parallèles a lieu. Ce problème concernant les accès concurrentiels à une ressource partagée telle qu'une variable globale s'accroît avec les logiciels multithread. Il existe de nombreuses utilisations valables des variables globales mais une utilisation erronée pourrait compliquer certaines situations de mise au point. Utilisez-les donc avec précaution.

Une fois le VI de variable globale enregistré, vous pouvez placer ses variables globales dans d'autres VIs en utilisant **Fonctions»Sélectionner un VI...** Si vous sélectionnez un VI de variable globale dans la boîte de dialogue de fichiers, le G place le nœud de la variable globale sur le diagramme. Vous pouvez également cloner, copier et coller, ou copier en faisant glisser une variable globale à partir de la fenêtre Hiérarchie.

Variables locales

Une variable locale vous permet d'écrire ou de lire une des commandes ou un des indicateurs de la face-avant de votre VI. L'écriture dans une variable locale revient au même que de transmettre des données à un terminal, excepté le fait que vous pouvez y écrire même si c'est une commande ou la lire même si c'est un indicateur. De même, le nombre de références des variables locales n'est pas limité pour une commande de face-avant donnée et certaines peuvent être en mode écriture et d'autres en mode lecture.

En fait, vous pouvez utiliser une commande de face-avant comme une entrée et une sortie grâce à une référence de variable locale.



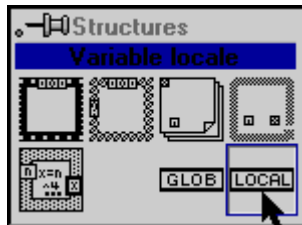
Remarque

Vous devez vous assurer que l'étiquette que vous choisissez pour votre variable locale est associée à un objet de la face-avant. Sinon, elle ne fonctionnera pas.

La manière la plus facile de créer une variable locale consiste à ouvrir le menu local de la commande de face-avant ou son terminal et de sélectionner **Créer»Variable locale**. Une variable locale s'affiche automatiquement sur le diagramme. Autrement, vous pouvez sélectionner la variable locale dans le menu **Fonctions»Structures**, illustré ci-dessous.

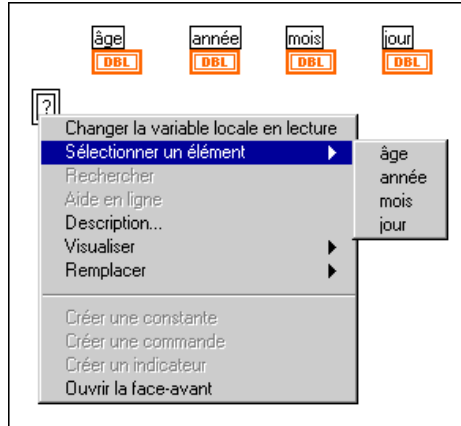
LOCAL

icône de variable locale

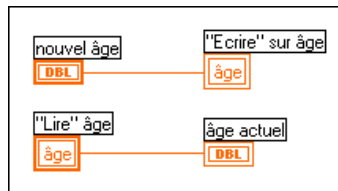


Nœud de variable locale

Un nœud semblable à une variable globale s'affiche. Vous pouvez ouvrir le menu local du nœud ou cliquer dessus avec l'outil Doigt pour sélectionner la commande que vous voulez lire, ou définir parmi une liste de commandes principales de face-avant, comme le montre l'illustration suivante. Vous pouvez également déterminer si vous voulez écrire ou lire la commande en sélectionnant **Changer la variable locale en écriture** ou **Changer la variable locale en lecture**.



L'illustration suivante montre comment plusieurs variables locales accèdent à la même commande, chacune ayant un sens différent, soit en lecture soit en écriture. La variable `âge` est utilisée deux fois dans le diagramme, une fois pour écrire et une autre pour lire.



Faites bien attention à ordonner l'accès aux variables locales ou variables globales afin d'obtenir les résultats escomptés, comme dans l'exemple précédent. Il se peut que "Ecrire" l'âge survienne avant "Lire" l'âge si vous ne créez pas la bonne séquence vous-même.

Partie IV

Sujets avancés

Cette section contient des informations sur les fonctions et techniques avancées du G que vous pouvez utiliser pour créer des instruments virtuels.

La quatrième partie, *Sujets avancés*, contient les chapitres suivants :

- Le chapitre 24, *Commandes personnalisées et définitions de type*, introduit les commandes personnalisées et les définitions de type.
- Le chapitre 25, *Appel de code d'autres langages*, décrit diverses méthodes pour appeler un code écrit dans un autre langage.
- Le chapitre 26, *Comprendre le système d'exécution du G*, décrit le fonctionnement en multitâche et l'exécution des VIs.
- Le chapitre 27, *Gestion de vos applications*, décrit comment gérer les fichiers dans vos applications en G.
- Le chapitre 28, *Performances*, est divisé en trois sections. La première section décrit le Profil de performance, une fonction qui vous donne des informations sur le temps d'exécution de vos VIs et surveille les applications qui fonctionnent en mono-threading, multi-threading et sur plusieurs processeurs. La deuxième section décrit les facteurs qui affectent la vitesse d'exécution. La troisième section décrit les facteurs qui affectent l'usage de mémoire.
- Le chapitre 29, *Portabilité et localisation*, aborde les problèmes concernant le port de VIs d'une plate-forme à une autre et la localisation des VIs.

Commandes personnalisées et définitions de type

Ce chapitre présente les commandes personnalisées et les définitions de type.

Vous pouvez personnaliser une commande ou un indicateur de face-avant pour l'adapter à votre application. Par exemple, vous voudrez peut-être qu'un interrupteur booléen affiche une valve fermée lorsqu'il est désactivé et une valve ouverte lorsqu'il est activé, ou encore qu'une commande curseur affiche son échelle à droite plutôt qu'à gauche, ou qu'une commande de type menu déroulant possède des éléments de texte ou de dessin prédéfinis.

Vous pouvez enregistrer une commande ou un indicateur personnalisé dans un répertoire ou une bibliothèque de VIs, exactement comme des VIs. Vous pouvez ensuite utiliser cette commande sur d'autres faces-avant. Vous pouvez également créer une icône pour votre commande personnalisée de manière à l'afficher accompagnée de son nom dans la palette de **Commandes**.

Si vous avez besoin de la même commande en différents endroits de vos VIs, vous pouvez créer une copie maître de cette commande, appelée définition de type. Lorsque vous apportez une modification à la définition de type, vous mettez automatiquement à jour tous les VIs qui l'utilisent.

Vous pouvez également utiliser une commande personnalisée sur un diagramme. Ainsi, vous pouvez créer une constante avec le même type de données que la commande. Si vous utilisez une définition de type sur un diagramme, la constante qui en résulte se met automatiquement à jour lorsque vous apportez une modification à la définition de type.

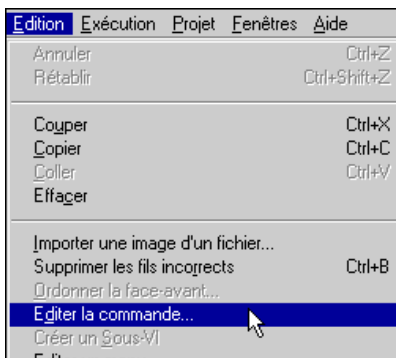
Les sections suivantes traitent de la manière de personnaliser les commandes.

Commandes personnalisées

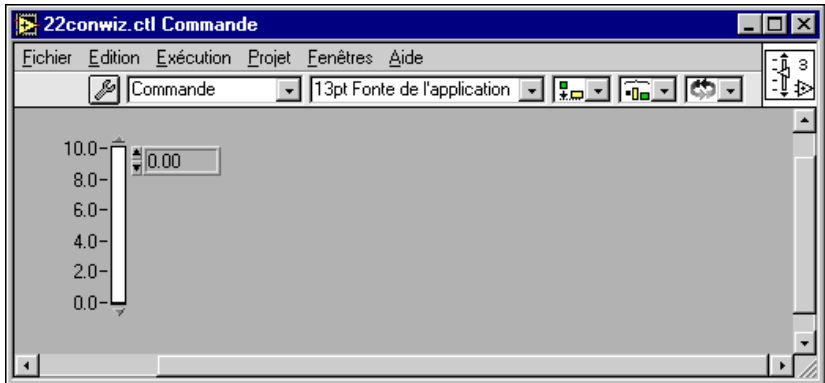
Utilisation de l'éditeur de commandes

Assurez-vous d'être en mode Edition lorsque vous souhaitez personnaliser une commande. Placez sur la face-avant une commande qui ressemble à celle que vous voulez créer. Par exemple, pour créer une glissière avec une échelle à droite, commencez par placer une glissière verticale sur la face-avant.

A l'aide de l'outil Flèche, sélectionnez la commande curseur et choisissez ensuite **Edition»Editer la commande...** Cette option n'est disponible que si vous sélectionnez une commande. Vous ne pouvez éditer qu'une seule commande de la face-avant à la fois.



Une fenêtre s'ouvre et affiche une copie de la commande. Cette fenêtre, présentée dans l'illustration suivante, s'appelle l'éditeur de commandes et est intitulée Commande *N*, qui est le nom affecté à la fenêtre de l'éditeur de commandes tant que vous n'avez pas affecté un nom définitif et enregistré la commande.



La fenêtre de l'éditeur de commandes ressemble à une face-avant, à la différence qu'elle n'est utilisée que pour l'édition et l'enregistrement d'une seule commande ; elle ne contient pas de diagramme et ne peut pas être exécutée.



Mode
Edition



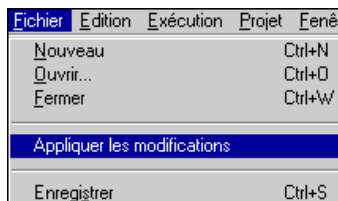
Mode
Personnalisé

L'éditeur de commandes possède un mode Edition et un mode Personnalisé. Le mode en cours est indiqué par un bouton de la barre d'outils, comme le montre l'illustration à gauche. L'éditeur de commandes est en mode Edition à sa première ouverture. En mode Edition, vous pouvez modifier la taille ou la couleur d'une commande et sélectionner les options de son menu local, comme en mode Edition d'une face-avant. En mode Personnalisé, vous pouvez modifier chaque composante d'une commande. Le mode personnalisé est détaillé plus loin dans ce chapitre.

Une fois la commande modifiée, vous pouvez l'utiliser à la place de la commande originale sur la face-avant sur laquelle vous avez ouvert l'éditeur de commandes. Vous pouvez également l'enregistrer et l'utiliser sur d'autres faces-avant.

Application des modifications d'une commande personnalisée

Lorsque vous êtes prêt à remplacer la commande originale de la face-avant par votre nouvelle commande personnalisée, sélectionnez **Fichier» Appliquer les modifications** dans le menu principal de l'éditeur de commandes.



Si votre face-avant originale est le seul endroit où vous allez utiliser la commande personnalisée, vous pouvez fermer la fenêtre de l'éditeur de commandes sans enregistrer la commande. Assurez-vous d'enregistrer le VI original avec la commande personnalisée afin de sauvegarder votre travail. Si vous voulez ultérieurement utiliser la commande personnalisée sur d'autres faces-avant, vous devez l'enregistrer en suivant les instructions de la section *Enregistrement des commandes personnalisées* de ce chapitre.

L'option **Appliquer les modifications** n'est disponible que si vous modifiez la commande. Cette option est désactivée s'il n'y a pas de commande originale à mettre à jour. C'est le cas lorsque vous supprimez ou remplacez la commande originale, fermez la face-avant originale ou ouvrez une commande personnalisée précédemment enregistrée en sélectionnant **Fichier»Ouvrir**.

Commandes personnalisées valides



Commande
invalide

Si l'éditeur de commandes contient plus d'une commande, le bouton **Commande invalide** s'affiche. Une commande personnalisée valide doit être une commande unique ou un cluster d'autres commandes. Le bouton **Commande invalide** s'affiche temporairement lorsque vous entrez ou sortez des commandes d'un cluster ou d'un tableau en les déplaçant. Pour obtenir une explication sur l'erreur, cliquez sur le bouton **Commande invalide**. Si l'éditeur de commandes contient plus d'une commande, le message suivant s'affiche.

Commandes supplémentaires sur la face-avant de la commande personnalisée.

Enregistrement des commandes personnalisées

Si vous voulez utiliser votre commande personnalisée sur d'autres faces-avant, choisissez **Fichier»Enregistrer sous...** dans le menu principal de l'éditeur de commandes. Une commande s'enregistre de manière identique à un VI, dans un répertoire ou une bibliothèque de VIs. Un répertoire ou une bibliothèque de VIs peut contenir des commandes, des VIs ou les deux.

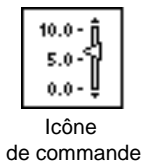
Si vous fermez la fenêtre de l'éditeur de commandes sans enregistrer vos modifications, une boîte de dialogue vous le signale et vous invite à l'enregistrer.

Utilisation des commandes personnalisées

Si vous enregistrez la commande personnalisée, vous pouvez l'utiliser sur d'autres faces-avant en sélectionnant **Commandes»Sélectionner une commande...** à partir de la face-avant de n'importe quel VI. Vous pouvez également l'utiliser sur des diagrammes en sélectionnant **Fonctions»Sélectionner un VI...** à partir du diagramme de n'importe quel VI. Si vous utilisez une commande personnalisée sur un diagramme, vous créez une constante avec un type de données identique à la commande personnalisée.

Pour plus d'informations sur l'ajout d'une commande personnalisée à la palette **Commandes**, consultez *Personnalisation des palettes Commandes et Fonctions* du chapitre 7, *Personnalisation de votre environnement*.

Création d'icônes



Si vous avez l'intention d'ajouter votre commande à la palette **Commandes** ou si la commande est une définition de type, créez une icône qui représente la commande avant de l'enregistrer. Pour cela, double-cliquez sur le carré de l'icône en haut à droite de la fenêtre de l'éditeur de commandes ou ouvrez son menu local et sélectionnez **Editer l'icône...** Cette icône représente la commande dans la palette **Commandes** ou dans la fenêtre Hiérarchie si la commande est une définition de type. Pour plus d'informations, consultez les sections *Définitions de type* de ce chapitre et *Utilisation de la fenêtre Hiérarchie* dans le chapitre 3, *Utilisation des sous-VIs*.

Instances indépendantes des commandes personnalisées

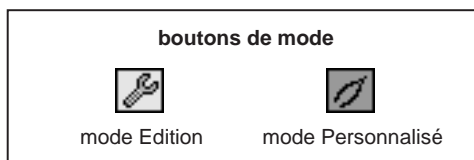
Vous pouvez ouvrir n'importe quelle commande personnalisée enregistrée en sélectionnant **Fichier»Ouvrir**. Une commande personnalisée sera toujours ouverte dans la fenêtre de l'éditeur de commandes.

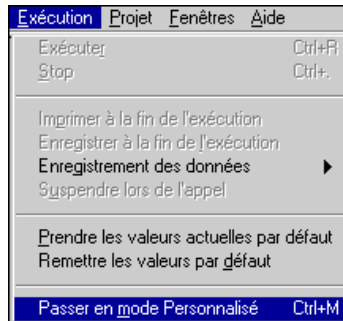
Les modifications apportées à la commande personnalisée une fois ouverte n'affectent pas les VIs qui l'utilisent. Lorsque vous utilisez une commande personnalisée sur une face-avant, il n'y a pas de relation entre cette instance de la commande personnalisée et le fichier ou la bibliothèque de VIs où elle a été enregistrée ; chaque instance est une copie séparée et indépendante.

Toutefois, vous pouvez créer une relation entre les instances de commande sur les différentes faces-avant de VI ou de diagrammes et la copie maître de la commande. Pour cela, vous devez enregistrer la commande personnalisée sous la forme d'une définition de type ou une définition de type stricte. Ensuite, toute modification apportée à la copie maître affectera toutes les instances de la commande dans tous les VIs qui l'utilisent. Pour plus d'informations, consultez la section *Définitions de type* plus loin dans ce chapitre.

Option Mode Personnalisé

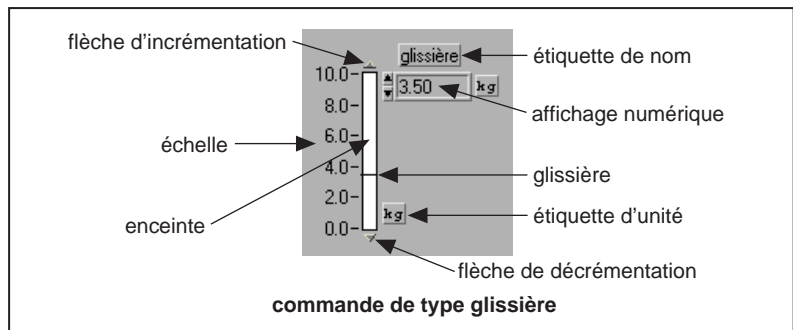
Vous pouvez apporter plus de modifications à une commande en mode Personnalisé de l'éditeur de commandes. Passez du mode Edition au mode Personnalisé en cliquant sur le bouton de mode de la barre d'outils de l'éditeur de commandes ou en sélectionnant **Passer en mode Personnalisé** ou **Passer en mode Edition** dans le menu **Exécution** lorsque l'éditeur de commandes est la fenêtre active, comme le montrent les illustrations suivantes.





Composantes indépendantes

Toutes les commandes sont conçues avec des composantes de plus petite taille. Par exemple, une commande curseur est composée d'une échelle, d'un emplacement, d'une glissière, de flèches d'incréméntation et de décrémentation, d'un afficheur numérique et d'une étiquette de nom. L'illustration suivante montre les différentes composantes d'une glissière.



Lorsque vous basculez en mode Personnalisé dans l'éditeur de commandes, les composantes de votre commande deviennent indépendantes. Vous pouvez modifier chaque composante sans affecter les autres. Par exemple, lorsque vous cliquez et faites glisser sur l'échelle de la glissière avec l'outil Flèche, seule l'échelle est déplacée. Vous pouvez sélectionner des composantes et les aligner ou les répartir en utilisant les menus déroulants **Aligner les objets** ou **Répartir les objets** dans la barre d'outils. Vous pouvez également modifier leur ordre de superposition en utilisant le menu déroulant **Réorganiser** de la barre d'outils. Le mode Personnalisé montre toutes les composantes de la commande, y compris celles qui sont masquées en mode Edition, telles que l'étiquette de nom ou la base d'une commande numérique.

Etant donné que les différentes composantes d'une commande sont détachées, vous ne pouvez pas exécuter ni modifier la valeur de la commande en mode Personnalisé. Remarquez que l'outil Doigt est désactivé. L'outil Bobine est toujours désactivé dans l'éditeur de commandes car vous n'utilisez pas de diagramme ou de commandes de connexion à un cadre connecteur.

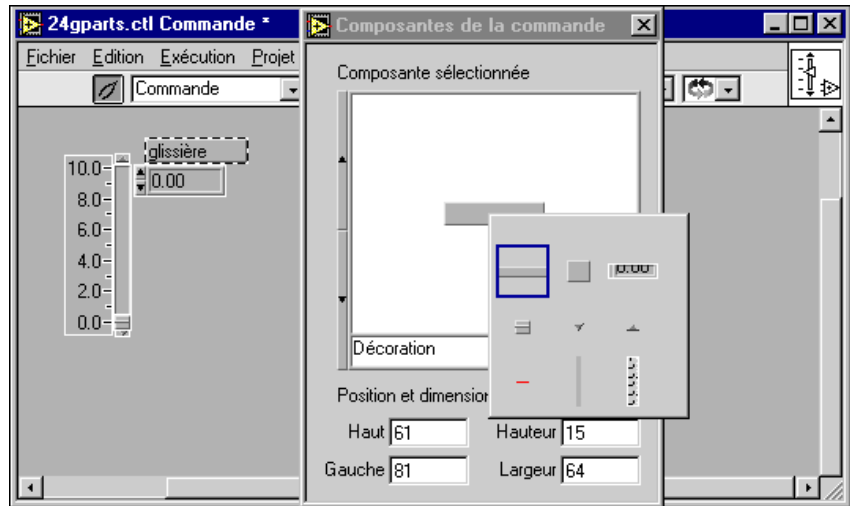
Fenêtre des composantes de l'éditeur de commandes

Afin de vous aider à dimensionner et positionner les composantes d'une commande, vous pouvez sélectionner **Fenêtres»Visualiser la fenêtre des composantes**. La fenêtre flottante qui s'affiche identifie les composantes de la commande et vous montre la position et la taille exactes de chaque composante. L'affichage *Composante sélectionnée* de la fenêtre des composantes montre l'image et le nom de la composante actuellement sélectionnée dans la fenêtre de l'éditeur de commandes. Vous pouvez voir un menu de toutes les composantes en cliquant sur l'affichage de la composante sélectionnée.

Vous pouvez également faire défiler les composantes de la commande en cliquant sur la flèche d'incrémentation ou de décrémentation de l'affichage de la composante sélectionnée. Lorsque vous modifiez cette composante, elle est sélectionnée sur la commande dans la fenêtre de l'éditeur de commandes. Lorsque vous sélectionnez, modifiez ou ouvrez un menu local sur une autre composante de la commande dans la fenêtre de l'éditeur de commandes, la composante qui se trouve dans l'affichage de la composante sélectionnée change également.

L'illustration suivante montre la fenêtre de l'éditeur de commandes à gauche recouverte par la fenêtre des composantes à droite. L'étiquette du nom de la glissière est la composante sélectionnée. La fenêtre des Composantes de la commande affiche le menu des composantes que vous voyez lorsque vous cliquez sur l'affichage de la composante sélectionnée.

Cet exemple montre que l'étiquette du nom est la composante sélectionnée mais que vous êtes sur le point de passer à la composante échelle.



La fenêtre des Composantes de la commande vous montre la position et la taille exactes de la composante sélectionnée (en pixels). Lorsque vous déplacez ou redimensionnez une composante dans l'éditeur de commandes, la position et la taille affichées dans la fenêtre des Composantes sont mises à jour. Vous pouvez également entrer les valeurs de position et de taille directement dans la fenêtre des Composantes de la commande pour déplacer ou redimensionner une composante dans l'éditeur de commandes. Cette fonction est utile lorsque vous devez coordonner la taille de deux composantes ou les aligner. Dans la précédente illustration, la fenêtre des Composantes affiche la position et la taille de l'étiquette de nom de la glissière – le coin supérieur gauche de l'étiquette se situe aux coordonnées (74, 84) en pixels et l'étiquette a une hauteur de 15 pixels par une largeur de 64 pixels.

La fenêtre des Composantes de la commande disparaît si vous basculez vers une autre fenêtre. Elle réapparaît lorsque vous retournez à l'éditeur de commandes.

Menus locaux du mode Personnalisé pour les différentes composantes

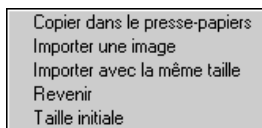
En mode Personnalisé, le menu local de la commande est remplacé par un menu pour chaque composante. Lorsque vous ouvrez le menu local d'une composante, vous voyez un menu avec certaines options en mode Edition et d'autres options en mode Personnalisé. Chaque composante possède un menu local différent. Les trois types de composantes de base que vous pouvez personnaliser sont :

- Les composantes de décoration, telles que l'emplacement de la glissière, la glissière et les flèches d'incrément et de décrémentation. Les composantes de décoration affichent une image.
- Les composantes de texte, telles que l'étiquette de nom d'une glissière. Elles sont composées d'une image d'arrière-plan (généralement un simple rectangle) et de texte.
- Des commandes en composantes, telles que la commande numérique utilisée pour l'afficheur numérique de la glissière. Les boutons rotatifs, les vu-mètres et les graphes déroulants utilisent également une commande numérique pour un affichage numérique. Certaines commandes sont encore plus compliquées que cela. Par exemple, le graphe utilise un tableau de clusters pour sa composante d'affichage du curseur.

Les sections suivantes décrivent de manière plus détaillée les différentes composantes et les options de menu local.

Composantes décoratives

Une composante décorative n'a pas d'interaction dynamique avec l'utilisateur. L'illustration suivante montre le menu local d'une composante décorative, telle qu'un emplacement de glissière. Pour ouvrir un menu local d'une composante décorative, vous devez être en mode Personnalisé. Vous devez ouvrir un menu local sur la composante elle-même et non pas sur l'image de la composante de la fenêtre des Composantes de la commande.



La liste suivante décrit les options disponibles dans le menu local.

- **Copier dans le presse-papiers** : place une copie de l'image de la composante dans le presse-papiers. Si vous sélectionnez **Copier dans le presse-papiers** pour l'emplacement de la glissière, le presse-papiers contiendra une image d'un rectangle incrusté long et étroit. Cette image du presse-papiers peut être collée sur n'importe quelle face-avant ou importée comme image dans une autre composante par le biais de l'option **Importer une image**. Ces images sont identiques aux **Décorations** de la palette **Commandes**.

Lorsque vous avez besoin de formes simples telles qu'un rectangle d'emplacement pour d'autres composantes, vous avez tout intérêt à utiliser les images copiées à partir d'autres composantes au lieu de les créer dans un programme de dessin. Les images extraites des composantes ou des décorations existantes conservent un meilleur aspect lors de leur redimensionnement. Par exemple, un rectangle créé dans un programme de dessin ne peut être agrandi que de manière uniforme en agrandissant sa superficie mais également l'épaisseur de sa bordure. Par contre, un rectangle copié à partir d'une autre composante telle que l'emplacement de la glissière conserve toujours la même bordure.

Un autre avantage réside dans le fait que les composantes intégrées s'affichent dans l'ensemble de manière identique sur les moniteurs couleur et les moniteurs noir et blanc.

De plus, vous pouvez colorer les images extraites des autres composantes ou décorations à l'aide de l'outil Pinceau. Les images importées à partir d'une autre source conservent leurs couleurs d'origine car ces couleurs sont une composante de la définition de cette image.

- **Importer une image** : remplace l'image en cours d'une composante décorative par l'image contenue dans le presse-papiers. Utilisez cette option afin de personnaliser l'apparence de chacune de vos commandes. Par exemple, vous pouvez importer des images qui représentent une valve ouverte ou fermée pour un interrupteur booléen. Si aucune image ne se trouve dans le presse-papiers, l'option **Importer une image** est désactivée.

Il existe un raccourci dans l'éditeur de commandes pour l'importation d'images dans une commande booléenne. Lorsque vous êtes en mode Edition, vous pouvez sélectionner **Importer une image»Vrai** ou **Importer une image»Faux** dans le menu local d'un booléen. Cette opération importe l'image à la fois dans l'état normal et dans l'état de transition correspondant. Pour plus d'informations sur les états de

transition, consultez la section *Composantes décoratives à plusieurs images* plus loin dans ce chapitre.

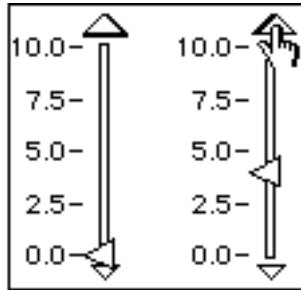
Vous pouvez également importer différentes images pour les états de transition en mode Personnalisé. Pour cela, ouvrez d'abord un menu local du bouton et utilisez **Élément image** afin de modifier la troisième image. Avec l'image **Vrai»Faux** dans le presse-papiers, ouvrez à nouveau un menu local et sélectionnez **Importer une image**. Répétez ces étapes pour la quatrième image (**Faux»Vrai**).

- **Importer avec la même taille** : remplace l'image actuelle mais conserve la taille originale de la composante en ajustant celle de l'image contenue dans le presse-papiers. Si aucune image ne se trouve dans le presse-papiers, l'option **Importer avec la même taille** est désactivée.
- **Revenir** : rétablit l'apparence originale de la composante. Cette option ne modifie pas la position de la composante. Si vous ouvrez la fenêtre de l'éditeur de commandes en sélectionnant **Éditer la commande** à partir d'une face-avant, l'éditeur rétablit la composante telle qu'elle apparaît sur la face-avant. Si vous ouvrez la fenêtre de l'éditeur de commandes en sélectionnant **Fichier»Ouvrir...**, l'option **Revenir** est désactivée.
- **Taille initiale** : rétablit la taille initiale d'une image de composante. Cette option est utile pour les images importées à partir d'autres applications et redimensionnées par la suite. Certaines de ces images perdent de leur qualité une fois redimensionnées. Vous pourrez ainsi rétablir leur taille initiale afin de les arranger. Si vous n'importez pas d'image, l'option **Taille initiale** est désactivée.

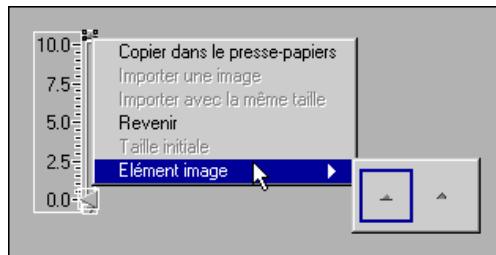
Composantes décoratives à plusieurs images

Certaines composantes décoratives possèdent plus d'une image qu'elles affichent à certains moments. Ces différentes images sont de même taille et de même couleur. Par exemple, la flèche d'incrémentement de la glissière est une image d'un triangle, normalement affichée en 3D. Elle possède également une autre image, un triangle encastré, qui s'affiche quand vous cliquez dessus avec l'outil Doigt pour incrémenter la valeur de la glissière.

L'illustration suivante montre deux images de flèches d'incrémentation en action.



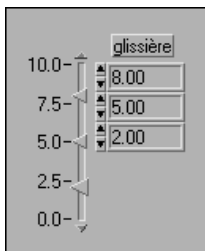
Une composante décorative à plusieurs images possède l'option **Élément image** dans son menu déroulant, comme le montre l'illustration suivante.



Cette option affiche toutes les images qui appartiennent à une composante décorative. L'élément image actuellement affiché possède une bordure noire. Lorsque vous importez une image, vous ne modifiez que l'élément image sélectionné. Pour importer une image pour un des autres éléments image, sélectionnez d'abord cet élément image et importez ensuite la nouvelle image.

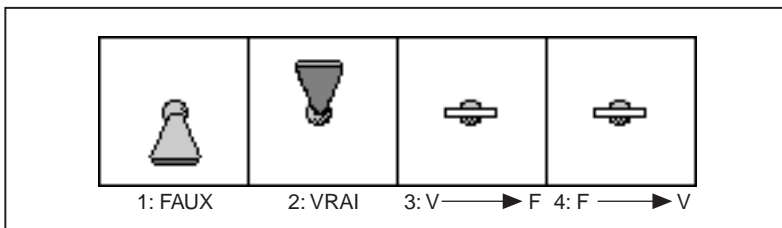
Composantes décoratives à images indépendantes

Une composante décorative avec plus d'une image peut posséder des images de différentes tailles et couleurs. La glissière, par exemple, utilise deux images de différentes tailles pour indiquer la glissière active sur une glissière à plusieurs valeurs. La glissière de l'exemple suivant utilise un plus gros triangle pour indiquer que la glissière du milieu est active.



Un interrupteur booléen possède également plusieurs images. Chaque image peut être de différentes taille et couleur. Un interrupteur booléen possède quatre différentes images : la première affiche l'état faux (FALSE) et la deuxième l'état vrai (TRUE). Vous pouvez utiliser les troisième et quatrième images lorsque vous définissez l'action mécanique d'une commande booléenne soit en **Commutation au relâchement** soit en **Armement au relâchement**.

Avec ces deux actions mécaniques, la valeur du booléen ne change pas tant que vous n'avez pas relâché le bouton de la souris. Entre le moment où vous cliquez sur le bouton et celui où vous le relâchez, le booléen affiche la troisième ou la quatrième image comme état transitoire. La troisième image concerne l'état transitoire du vrai au faux et la quatrième du faux au vrai. Dans l'illustration suivante d'un interrupteur à bascule, la troisième et la quatrième image sont identiques, mais ce n'est pas toujours le cas.



Lorsqu'une composante décorative peut contenir différentes tailles d'images, elle possède l'option **Tailles indépendantes** dans son menu déroulant, comme le montre l'illustration suivante.



Cette option ne peut être activée que lorsque vous êtes en mode Personnalisé. Elle vous servira à déplacer et à redimensionner chaque image sans modifier les autres images de la composante décorative. Généralement, cette option n'est pas activée et, lorsque vous déplacez ou redimensionnez l'image en cours de la composante décorative, ses autres images bougent également de manière identique ou changent de taille proportionnellement.

Composantes de texte

Une composante de texte est une image qui contient du texte. Le menu local d'une composante de texte, telle qu'une étiquette de nom, possède certains éléments identiques à ceux du menu local d'une composante décorative. Les autres éléments de ce menu sont les mêmes que ceux du menu local du texte en mode Edition de la face-avant. L'illustration suivante montre un exemple de menu local d'une composante de texte.



Seule l'image d'arrière-plan d'une composante de texte est affichée dans la fenêtre des composantes, non pas le texte lui-même. L'image d'arrière-plan peut être personnalisée, pas le texte.

Commandes composantes

Une commande peut inclure d'autres commandes composantes. L'afficheur numérique d'une glissière, d'un bouton rotatif, d'un vu-mètre ou d'un graphe déroulant constitue un exemple commun. Il n'existe aucune différence entre l'afficheur numérique et la commande numérique ordinaire d'une face-avant excepté le fait que l'afficheur numérique sert de composante d'une autre commande.

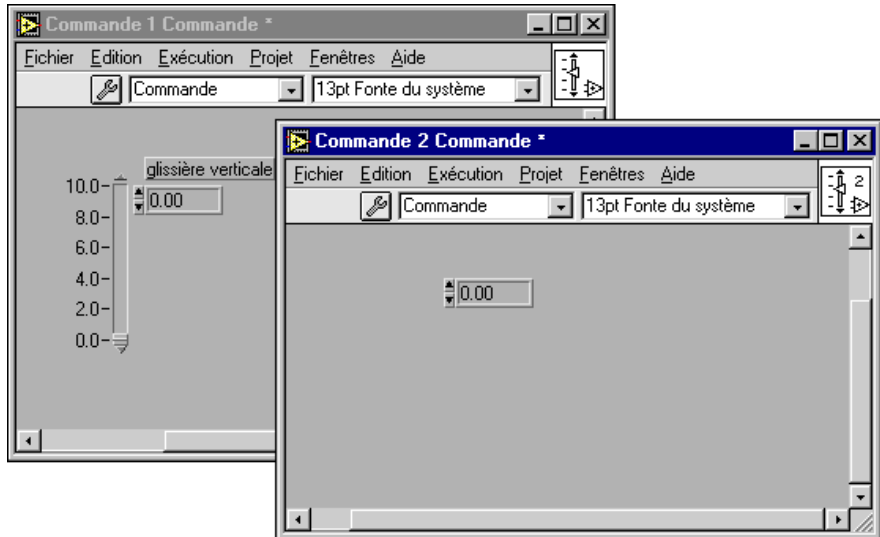
L'afficheur numérique est également constitué de composantes. Lorsque vous éditez la commande originale dans l'éditeur de commandes, l'afficheur numérique se comporte comme une composante unique. Vous ne pouvez donc pas modifier ou déplacer ses composantes de manière individuelle. Toutefois, vous pouvez ouvrir l'éditeur de commandes pour l'afficheur numérique pour le personnaliser.

Pour personnaliser une commande qui est une composante d'une autre commande, ouvrez l'éditeur de commandes de cette commande. Vous pouvez directement ouvrir la fenêtre de l'éditeur de commandes pour la composante à partir de la face-avant initiale, si elle peut être sélectionnée de manière séparée à la commande principale en mode Edition. En effet, l'afficheur numérique peut être sélectionné séparément de la commande curseur par exemple. Ensuite, vous pouvez choisir **Edition»Editer la commande...**

Vous pouvez toujours ouvrir la fenêtre de l'éditeur de commandes pour la composante à partir de la fenêtre de l'éditeur de commandes de la commande principale, si elle est en mode personnalisé. Sélectionnez la composante dans l'éditeur de commandes et choisissez **Edition»Editer la commande...** Les éditeurs de commandes peuvent être indéfiniment imbriqués de cette manière, mais la plupart des commandes n'utilisent d'autres commandes composantes qu'au niveau principal. Le graphe constitue une exception car il utilise des commandes compliquées constituées de composantes, qui à leur tour utilisent d'autres commandes constituées de composantes.

Vous ne pouvez pas ouvrir une deuxième fenêtre d'éditeur de commandes pour la commande principale qui est déjà en cours de personnalisation.

L'illustration suivante montre l'éditeur de commandes pour la glissière à gauche et une fenêtre de l'éditeur de commandes pour l'afficheur numérique à droite. Vous n'avez pas besoin d'être en mode Personnalisé pour ouvrir une fenêtre imbriquée dans l'éditeur de commandes sauf si vous êtes dans l'impossibilité de sélectionner la composante de la commande en mode Edition.



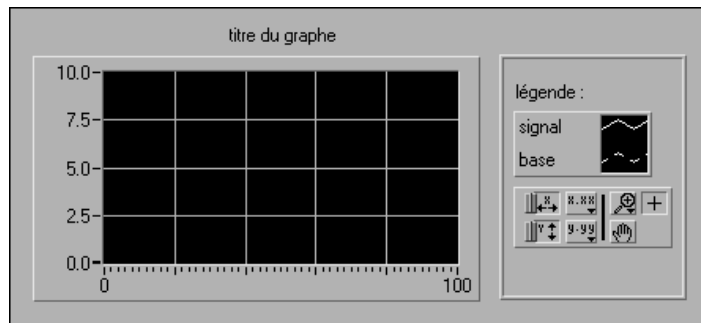
Ajout de composantes décoratives à des commandes personnalisées

Lorsque vous créez une commande personnalisée dans l'éditeur de commandes, vous pouvez apporter encore plus de modifications à son apparence en lui ajoutant des composantes décoratives ou de texte.

Si vous collez une image ou du texte à partir du presse-papiers, créez une étiquette avec l'outil Texte ou sélectionnez une image à partir de **Commandes»Décorations**. Cette image ou ce texte deviennent une composante de votre commande et s'affichent avec la commande lorsque vous la placez sur la face-avant. Vous pouvez effectuer cette opération soit en mode Edition soit en mode Personnalisé dans l'éditeur de commandes. Vous pouvez déplacer, redimensionner ou modifier l'ordre de superposition de la nouvelle composante, exactement comme avec une autre composante. Votre ajout s'affiche comme une composante de décoration dans la fenêtre des Composantes de la commande en mode Personnalisé.

Vous pouvez également supprimer des composantes de décoration lorsque vous vous trouvez dans l'éditeur de commandes.

L'illustration suivante montre un graphe personnalisé avec des composants de décoration, dont les étiquettes *Titre du graphe*, *Légende* : et un cadre autour des composants de légende.



Lorsque vous utilisez une commande personnalisée sur d'autres faces-avant, vous pouvez modifier la taille de n'importe quelle composante de décoration que vous ajoutez, mais vous ne pouvez pas les déplacer.

Avertissements à propos des commandes personnalisées

Vous devez faire attention à plusieurs choses lorsque vous créez des commandes personnalisées.

- Les images créées sur une plate-forme changent légèrement d'aspect une fois chargées sur une autre plate-forme (ceci s'applique aux images importées dans un menu déroulant d'images ou utilisées en arrière-plan de face-avant). Par exemple, une image de forme irrégulière ou un arrière-plan transparent pourraient avoir un arrière-plan blanc sur une autre plate-forme. Consultez la section *Différences entre les images* dans le chapitre 29, *Portabilité et localisation*.
- L'éditeur de commandes ne peut modifier que l'apparence d'une commande, et non pas son comportement. Ainsi :
 - Vous ne pouvez pas modifier la manière dont une commande affiche ses données.
 - Vous ne pouvez pas modifier la manière dont une commande se comporte lorsque vous l'éditez, surtout si vous la redimensionnez.

Par exemple, lorsque vous allongez une commande de type menu déroulant, les flèches d'incrémentatation et de décrémentatation deviennent également plus hautes. Si vous déplacez les flèches d'incrémentatation et de décrémentatation pour qu'elles soient côte à côte au bas d'un menu déroulant, ce dernier continue à les allonger lorsqu'il lui-même s'allonge et vous obtenez un résultat inattendu.

- Les commandes personnalisées restent souvent correctes mais se comportent parfois de manière étrange. Si vous souhaitez que la commande demeure telle que vous l'avez créée mais que vous n'êtes pas satisfait du comportement d'édition irrégulier, lisez la section suivante à propos des définitions de type strict, *Définitions de type*, pour apprendre à restreindre l'édition.

Définitions de type

Une *définition de type* est une copie maître d'une commande. Vous devez utiliser l'éditeur de commandes pour créer la copie maître, ou définition de type. Les définitions de type sont utiles lorsque vous utilisez le même type de commande dans plusieurs VIs. Vous pouvez enregistrer la commande sous forme de définition de type et l'utiliser dans tous vos VIs. Ainsi, si vous modifiez cette commande, vous pouvez mettre à jour le fichier de la définition de type au lieu de devoir modifier toutes les commandes des VIs qui l'utilisent.

Définition de type général : correspondance des types de données

Une définition de type oblige le type de données de la commande à être identique quel que soit l'endroit où il est utilisé. Utilisez une définition de type lorsque vous voulez utiliser une commande du même type de données à plusieurs endroits et que vous souhaitez automatiquement modifier ce type de données quel que soit l'endroit où il est utilisé. Par exemple, supposons que vous créez une définition de type à partir d'une commande numérique à double précision. Par la suite, vous pouvez utiliser cette définition de type dans différents VIs. Supposez que vous modifiez ensuite la définition de type en une commande numérique d'entier 16 bits. Lorsque vous modifiez la définition de type, vous pouvez automatiquement mettre à jour tous les VIs qui utilisent cette définition de type. Seules les instances de définition de type que vous définissez spécifiquement de manière à ne pas les mettre à jour automatiquement, ne le sont pas. Pour plus d'informations, consultez la section *Utilisation des définitions de type* plus loin dans ce chapitre.

Vous pouvez également créer une définition de type sous forme de cluster, tel qu'un cluster de deux entiers et une chaîne de caractères. Si vous modifiez cette définition de type en un cluster de deux entiers et de deux chaînes de caractères, vous pouvez mettre à jour la définition de type quel que soit l'endroit où elle est utilisée.

Tant que le type de données correspond à la copie maître, une définition de type peut posséder un nom, une description, une valeur par défaut, une

taille, une couleur, ou même un style de commande différents (par exemple, un bouton rotatif au lieu d'une glissière).

Définition de type strict : tout doit correspondre

Une définition de type peut également obliger pratiquement *tout* ce qui concerne la commande à être identique partout, non seulement son type de données mais également sa taille, sa couleur et son apparence. C'est ce qu'on appelle une *définition de type strict*.

Les seuls aspects d'une commande qui peuvent différer d'une copie maître de définition de type strict sont le nom, la description et la valeur par défaut. Par exemple, supposons que vous créez une définition de type strict qui est une commande numérique à double précision avec un cadre rouge. Comme la définition de type général, si vous modifiez la définition de type strict en un entier, vous mettez automatiquement à jour tous les VIs qui l'utilisent. Toutefois, contrairement à la définition de type général, les autres modifications telles que le passage de la couleur rouge en bleu, nécessitent également une mise à jour des VIs qui utilisent cette définition de type strict. De plus, vous ne pouvez pas désactiver la mise à jour automatique d'une instance de définition de type strict.



Remarque

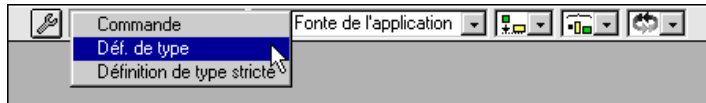
De nombreux attribut nodes ne sont pas disponibles pour les définitions de type strict. Seuls ceux qui affectent l'apparence de la commande, comme Visible, Désactivé, Focus de la touche, Clignotement, Position et Limites sont disponibles.

Définitions de type sur le diagramme

Lorsque vous utilisez une définition de type sur un diagramme, elle prend toujours l'aspect d'une constante, jamais celui d'une commande ni d'un indicateur. Par conséquent, une instance de définition de type strict sur un diagramme agit comme une instance de définition de type général. Elle ne se met automatiquement à jour que lorsque le type de données de la définition de type strict change.

Création de définitions de type

Vous pouvez créer une définition de type en utilisant le menu déroulant de la barre d'outils d'une fenêtre de l'éditeur de commandes, comme le montre l'illustration suivante. Créez librement la commande et choisissez **Fichier>Enregistrer** dans la fenêtre de l'éditeur de commandes.



Vous pouvez ouvrir n'importe quelle définition de type enregistrée en sélectionnant **Fichier»Ouvrir...** Une définition de type s'ouvre toujours dans une fenêtre de l'éditeur de commandes. Toute modification apportée à une définition de type affecte tous les VIs qui l'utilisent.

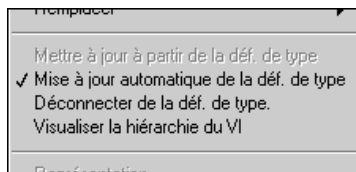
Utilisation des définitions de type

Placez les définitions de type général et les définitions de type strict sur la face-avant ou le diagramme d'un VI de la même manière que les commandes personnalisées. Vous pouvez éditer et exécuter une définition de type sur votre face-avant ou diagramme comme n'importe quelle commande ou constante.

Remarque

Vous ne pouvez pas éditer une définition de type strict sur votre face-avant excepté pour modifier son nom, sa description ou sa valeur par défaut.

Vous pouvez reconnaître qu'une commande est une définition de type à partir des éléments de son menu local, comme le montre l'illustration suivante. Une définition de type sur votre face-avant ou diagramme peut être reconnue du fait que vous ne pouvez pas l'éditer et aussi à cause de l'absence de la plupart des options de son menu local.



Pour chaque définition de type que vous utilisez sur une face-avant ou un diagramme, le VI conserve une connexion à la bibliothèque de VIs ou au fichier dans lequel elle a été enregistrée. Vous pouvez voir cette connexion en action si vous placez une définition de type sur une face-avant ou un diagramme, puis la sélectionnez et choisissez **Edition»Editer la commande...** L'éditeur de commandes qui s'ouvre est la définition de type que vous avez enregistrée, avec le nom que vous lui avez affecté au lieu de la Commande *N* générale.

Mise à jour des définitions de type

Votre environnement de développement G s'assure que le type de données reste identique partout où la définition de type est utilisée. Il s'assure également que tout ce qui concerne une définition de type strict reste identique dans toutes les faces-avant où il est utilisé. Vous pouvez automatiquement corriger une définition de type général ou une définition de type strict sur votre face-avant. Vous pouvez également remplacer celles qui sont incorrectes sur la face-avant par une copie exacte enregistrée dans le fichier ou la bibliothèque de VIs et celles sur le diagramme par une constante dont le type de données correspond à la commande enregistrée.

Si vous éditez une instance de définition de type sur votre face-avant, en modifiant sa couleur et en la redimensionnant, vous ne voudrez peut-être pas que la fonction de mise à jour automatique soit activée. Vous pouvez ouvrir le menu local de la définition de type de votre face-avant et désactiver l'option **Mise à jour automatique de la déf. de type**. Au lieu d'effectuer une mise à jour automatique de cette définition de type, le VI affiche une flèche brisée et la définition de type sur la face-avant est désactivée. Vous ne pouvez pas exécuter le VI tant que vous n'avez pas réparé la définition de type, soit en sélectionnant l'option **Mettre à jour à partir de la déf. de type** à partir du menu local, soit en modifiant le type de données afin qu'il corresponde à la définition de type. L'option **Mise à jour automatique de la déf. de type** n'est pas disponible dans le menu local d'une définition de type strict puisqu'elle se met toujours automatiquement à jour.

Lorsque vous utilisez une définition de type, vous pouvez lui attribuer une autre valeur par défaut. Toutefois, si le type de données de la définition de type change, toutes les données par défaut sont mises à jour à partir de la copie maître mais les anciennes valeurs par défaut ne peuvent pas être converties dans le nouveau type de données. C'est ce qui arrive lors du remplacement d'un nombre par une chaîne de caractères. Sinon, chaque valeur par défaut est conservée.

Si vous câblez une définition de type en utilisant **Créer une constante**, **Créer une commande** ou **Créer un indicateur**, la définition de type est mise à jour à partir de la copie maître. Pour plus d'informations sur **Créer une constante**, **Créer une commande** ou **Créer un indicateur**, consultez le chapitre 17, *Introduction au diagramme*.

Recherche des définitions de type

Un VI devant conserver une connexion avec chaque définition de type, le fichier ou la bibliothèque de VIs qui contient la définition de type doit être disponible pour exécuter un VI qui l'utilise. Si vous ouvrez un VI et, si une définition de type dont le VI a besoin n'est pas trouvée, les instances de cette définition de type dans le VI sont désactivées et la flèche d'exécution est brisée. Pour résoudre le problème, vous devez trouver et ouvrir la bonne définition de type ou ouvrir un menu local sur l'instance désactivée et sélectionner **Déconnecter de la déf. de type**. La déconnexion de la définition de type supprime les restrictions du type de données de l'instance, ce qui la transforme en commande ou constante ordinaire. Vous ne pouvez pas rétablir cette connexion tant que vous ne trouvez pas la définition de type et que vous ne remplacez pas la commande par cette dernière.

Définitions de type cluster

Si vous utilisez une définition de type ou une définition de type strict sous la forme d'un cluster, il vaut mieux utiliser les fonctions "Assembler par nom" et "Séparer par nom" sur le diagramme afin d'accéder aux éléments du cluster au lieu des fonctions Assembler et Séparer. Ces fonctions font référence à des éléments du cluster par leur nom plutôt que par leur ordre dans le cluster. Elles ne sont pas affectées lorsque vous réorganisez les éléments ou que vous ajoutez de nouveaux éléments à la définition de type cluster. Si vous supprimez un élément auquel vous faites référence par "Assembler par nom" et "Séparer par nom", vous devez modifier votre diagramme. Pour plus d'informations, consultez la description de ces fonctions dans le chapitre 14, *Commandes et indicateurs de tableau et de cluster*.

Appel de code d'autres langages

Ce chapitre présente les différentes méthodes d'appel de code écrit dans d'autres langages. Ces méthodes comprennent l'utilisation des protocoles spécifiques aux plates-formes, la création des Code Interface Node pour appeler un code écrit spécifiquement pour des VIs et l'utilisation du nœud "Appeler une fonction" pour appeler des Bibliothèques de liaisons dynamiques (fichiers .DLL) sous Windows, des Fragments de code sur Macintosh et des Bibliothèques partagées sur UNIX. Une autre méthode consiste à utiliser le convertisseur de Panneau de fonction LabWindows/CVI afin de convertir un driver d'instrument écrit sous LabWindows/CVI.

Exécution d'autres applications à partir de vos VIs

Vous pouvez exécuter d'autres applications à partir de vos VIs. Les méthodes sous Windows et UNIX sont différentes de celles de Macintosh.

(Windows, UNIX) Vous devez utiliser le VI System Exec pour exécuter d'autres applications à partir de vos VIs. Vous pouvez ouvrir ce VI dans la palette **Fonctions»Communication** pour exécuter une ligne de commande à partir de votre VI. La ligne de commande peut comprendre tous les paramètres supportés par l'application que vous prévoyez de lancer.

Si vous pouvez accéder à l'application par le biais de TCP/IP (ou DDE sous Windows), vous pouvez transmettre des données ou des commandes à l'application. Consultez la documentation de l'application que vous prévoyez d'utiliser pour connaître l'étendue de ses capacités de communication. Si vous êtes un utilisateur de LabVIEW, vous pouvez également consulter le chapitre 20, *Introduction à la communication* et le chapitre 21, *TCP et UDP* du *Manuel de l'utilisateur LabVIEW* pour plus d'informations sur les techniques d'utilisation des VIs de réseau afin de transférer des informations aux autres applications.

(Macintosh) Vous devez utiliser des VIs Apple Event pour exécuter d'autres applications à partir de vos VIs. Apple Event est un protocole spécifique au

Macintosh par le biais duquel des applications communiquent entre elles. Il peut être utilisé pour envoyer des commandes entre les applications. Vous pouvez également l'utiliser pour lancer d'autres applications. Si vous êtes un utilisateur de LabVIEW, consultez le chapitre 21, *TCP et UDP* du *Manuel de l'utilisateur LabVIEW* pour plus de détails sur les différentes méthodes d'utilisation des VIs Apple Event dans le G afin de lancer ou de contrôler d'autres applications.

Utilisation du nœud “Appeler une fonction d'une DLL”

Vous pouvez appeler la plupart des bibliothèques partagées standard (sous Windows, ce sont des Bibliothèques de liaisons dynamiques, ou DLL, sur Macintosh ce sont des Fragments de code et sur UNIX des bibliothèques partagées) en utilisant le nœud “Appeler une fonction”. Ce dernier comprend un grand nombre de types de données et de conventions d'appel. Vous pouvez l'utiliser pour appeler des fonctions à partir de la plupart des bibliothèques standard et personnalisées.

Le nœud “Appeler une fonction” est approprié lorsque vous avez un code existant que vous voulez appeler ou, si vous êtes familier avec la procédure de création d'une DLL sous Windows, des Fragments de code sur Macintosh ou des bibliothèques partagées sur UNIX. Etant donné qu'une bibliothèque utilise un format standard dans de nombreux environnements de développement, vous pouvez pratiquement utiliser n'importe quel environnement de développement afin de créer une bibliothèque qui puisse être appelée par le G. Consultez la documentation de votre compilateur pour savoir s'il peut créer des bibliothèques partagées standard.

Pour obtenir une description détaillée de ce nœud, consultez la section *Appeler une fonction d'une* plus loin dans ce chapitre.

Sur un système d'exploitation supportant le multithreading, vous pouvez simultanément effectuer plusieurs appels à une bibliothèque de liaison dynamique (fichier .DLL) ou à une bibliothèque partagée. Par défaut, tous les nœuds d'appel de bibliothèque, y compris les nœuds des précédentes versions, s'exécutent dans le thread de l'interface utilisateur. Avant de configurer un nœud de bibliothèque d'appel comme ré-entrant, assurez-vous que la fonction appelée peut être exécutée simultanément par plusieurs threads. Pour plus d'informations, consultez la section *Multithreading* du chapitre 26, *Comprendre le système d'exécution du G*.

Utilisation des Code Interface Nodes

Pour les applications qui nécessitent les meilleures performances possibles, ou si vous voulez transmettre des structures de données arbitraires en code C, vous pouvez créer un CIN (Code Interface Node). Par le biais des CIN, vous pouvez appeler un code écrit spécifiquement pour des VIs en langage G.

Le CIN est une méthode commune d'appel du code C à partir du G. Vous pouvez arbitrairement entrer ou sortir des structures de données complexes d'un CIN. Dans certains cas, vous obtiendrez de meilleures performances en utilisant des CIN, les structures de données étant transmises au CIN au même format que leur stockage dans le G.

Toutefois, pour obtenir ce niveau de performance, vous devez apprendre à créer un CIN. Pour cela, vous devez être un bon développeur C et disposer de suffisamment de temps pour créer le CIN souhaité. De même, étant donné que les CIN sont étroitement liés au G, vous ne pourrez pas utiliser n'importe quel compilateur.

Si vous êtes un utilisateur de LabVIEW, vous pouvez consulter le *LabVIEW Code Interface Reference Manual*, disponible au format PDF sur les disquettes ou CD du logiciel, pour plus d'informations sur la création des Code Interface Nodes.

Appeler une fonction d'une

“Appeler une fonction d'une DLL” vous permet d'appeler directement une DLL 16 bits sous Windows 3.1, une DLL 32 bits sous Windows 95/NT, un Fragment de code sur Macintosh ou une fonction de bibliothèque partagée sur UNIX.

(Macintosh) “Appeler une fonction d'une DLL” utilise le Code Fragment Manager (CFM) du Macintosh. C'est un standard sur toutes les machines PowerMac. Les ordinateurs Macintosh 680x0 utilisent l'extension CFM. De plus, le nœud “Appeler une fonction” sur les ordinateurs Macintosh 680x0 ne peut pas appeler des fonctions à argument variable. Les bibliothèques partagées du Macintosh s'exécutent différemment par rapport aux autres plates-formes. Un fichier peut contenir plus d'un fragment de code, chacun ayant un nom.

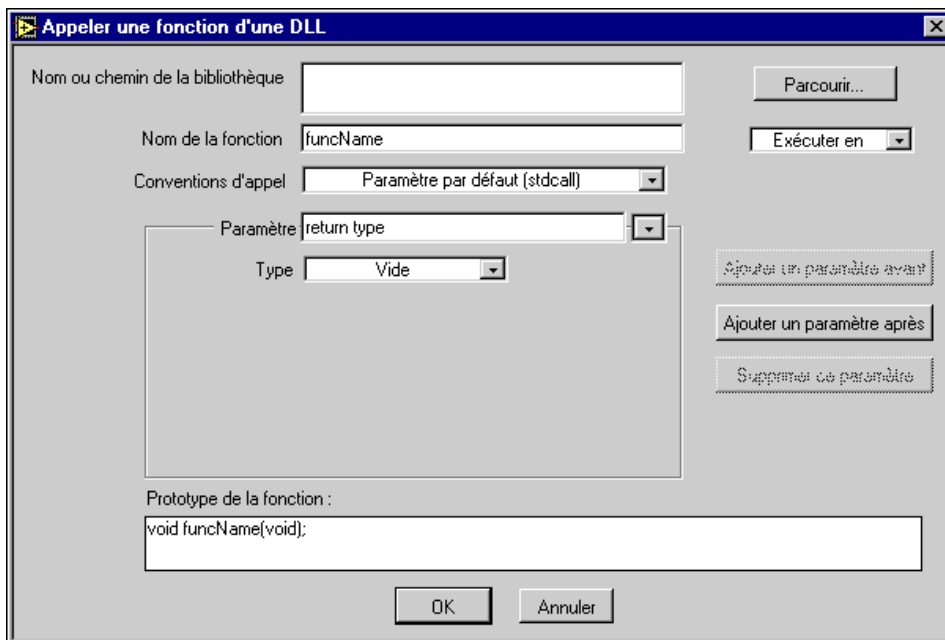


“Appeler une fonction”, illustré à gauche, se trouve dans **Fonctions» Avancé**.

Si vous double-cliquez sur “Appeler une fonction d’une DLL” ou sélectionnez **Configurer...** dans son menu local, le G affiche une boîte de dialogue que vous pouvez utiliser afin de spécifier la bibliothèque, la fonction, les paramètres et la valeur retournée du nœud ainsi que les conventions d’appel sous Windows. Lorsque vous cliquez sur le bouton **OK**, le nœud ajoute automatiquement le bon nombre de terminaux et définit les bons types de données des terminaux.

La valeur transmise par la fonction est retournée au terminal de droite des deux terminaux supérieurs du nœud. Si aucune valeur n’est retournée, ces deux terminaux ne sont pas utilisés. Chaque paire de terminaux supplémentaire correspond à un paramètre de la liste de paramètres des fonctions. Vous pouvez transmettre une valeur à la fonction en vous connectant au terminal de gauche de la paire de terminaux. Vous pouvez lire la valeur d’un paramètre à la suite de l’appel d’une fonction en vous connectant au terminal de droite de la paire de terminaux.

L’illustration suivante montre la boîte de dialogue “Appeler une fonction d’une DLL”.



Au fur et à mesure que vous sélectionnez des éléments dans la boîte de dialogue, un indicateur, appelé Prototypé de la fonction, affiche le prototype C de la fonction sélectionnée.

**Remarque**

L'appel des bibliothèques partagées écrites et compilées en C++ n'a pas été testé et peut ne pas fonctionner.

**Remarque**

Sous Windows 3.1, un maximum de 29 arguments de 4 octets peuvent être transmis en utilisant "Appeler une fonction". Les paramètres à virgule flottante double précision passés aux valeurs équivalent à 8 octets et comptent pour deux arguments. Par conséquent, vous êtes limité à 14 paramètres à virgule flottante double précision passés par valeur.

**Remarque**

Sous Windows 3.1, vous ne pouvez pas avoir plus de deux nœuds d'appel de bibliothèque en mémoire qui appellent la même fonction à l'intérieur de la même DLL avec différents arguments.

**Remarque**

Sous Windows 3.1, les DLL doivent être en 16 bits. Sous Windows 95/NT, les DLL doivent être en 32 bits. Si vous avez une DLL 16 bits que vous voulez appeler à partir de Windows 95/NT, vous devez soit la recompiler en DLL 32 bits, soit créer une DLL "thunking". Pour plus d'informations sur le thunking des DLL, consultez la documentation de Microsoft.

Conventions d'appel (Windows)

Utilisez les conventions d'appel du menu déroulant pour sélectionner celles de la fonction. La convention d'appel par défaut pour Windows 3.1 est Pascal et Stdcall pour Windows 95/NT. Ceci correspond à la convention d'appel utilisée par la plupart des DLL. Une autre option consiste à utiliser des conventions d'appel C. Consultez la documentation de la DLL que vous essayez d'appeler pour connaître les conventions d'appel appropriées.

Listes des paramètres

Au départ, "Appeler une fonction d'une DLL" n'a pas de paramètre et sa valeur retournée est vide. Vous pouvez cliquer sur les boutons **Ajouter un paramètre avant** et **Ajouter un paramètre après** pour ajouter des paramètres à la fonction. Vous pouvez cliquer sur le bouton **Supprimer ce paramètre** pour supprimer un paramètre.

Vous pouvez utiliser le menu déroulant de paramètres pour sélectionner les différents paramètres ou la valeur retournée. Une fois sélectionné, vous pouvez modifier le nom du paramètre pour qu'il soit plus descriptif. Le nom du paramètre n'affecte pas l'appel, mais il est propagé aux fils de liaison de sortie. Des noms descriptifs facilitent le passage d'un paramètre à un autre.

Indiquez le type de chaque paramètre en utilisant le menu déroulant des types. Le type de retour se limite soit à **Vide**, ce qui signifie que la fonction ne retourne pas de valeur, soit à **Numérique**, soit à **Chaîne de caractères**.

Pour les paramètres, vous pouvez sélectionner **Numérique**, **Tableau**, **Chaîne de caractères** ou **Adapter au type**.

Lorsque vous sélectionnez un élément du menu déroulant des types, vous apercevez des éléments supplémentaires que vous pouvez utiliser pour indiquer des détails concernant le type de données et la manière de passer les données à la fonction de bibliothèque. Le **Nœud Bibliothèque d'appel** possède un certain nombre d'éléments pour les types de données car les types de données requis par les différentes bibliothèques sont très variés. Consultez la documentation de la bibliothèque que vous appelez afin de déterminer les types de données à utiliser.

- **Vide** : le type `vide` n'est accepté que pour la valeur retournée. Cet élément n'est pas disponible pour les paramètres. Utilisez-le pour la valeur retournée si votre fonction ne retourne pas de valeur.
- **Numériques** : pour les types de données numériques, vous devez indiquer le type numérique exact en utilisant le menu déroulant des types de données. Les éléments sont les suivants :
 - Versions signées et non signées des entiers 8 bits, 16 bits et 32 bits
 - Nombres simple précision de quatre octets
 - Nombres double précision de huit octets

Vous ne pouvez pas utiliser des nombres à précision étendue ni des nombres complexes. Ils ne sont généralement pas utilisés dans les bibliothèques standard.

Vous devez également utiliser le menu déroulant des formats pour indiquer si vous voulez transmettre la valeur ou un pointeur sur la valeur.



Remarque

Sous Windows 3.1, vous ne pouvez pas utiliser les types de données simple précision ou double précision comme type de retour. Il n'existe pas de méthode standard de retour des nombres simple précision et double précision des DLL. Les valeurs à virgule flottante retournées sont différemment implémentées par chaque compilateur. Si vous devez retourner un nombre simple précision ou double précision, transmettez les données comme paramètre plutôt que comme valeur retournée.

- **Tableaux** : vous pouvez indiquer le type de données des tableaux (en utilisant les mêmes éléments que ceux des types de données numériques), le nombre de dimensions et le format à utiliser dans la

transmission au tableau. Utilisez l'élément **Format** pour choisir si vous voulez transmettre un pointeur aux données du tableau ou un **handle sur tableau** qui est un pointeur d'un pointeur vers une valeur de quatre octets pour chaque dimension suivi par les données. Si vous sélectionnez **Pointeur sur les données du tableau**, passez la dimension du tableau comme paramètre(s) séparé(s).



Remarque

Sous Windows 3.1, le nœud d'appel de bibliothèque n'utilise pour les tableaux que le format de pointeur sur les données du tableau. De plus, sous Windows 3.1 vous pouvez indiquer le passage des données en utilisant un pointeur de type Huge. Vous pouvez utiliser un pointeur de type Huge si vous devez passer plus de 64 Ko de données dans le tableau. N'activez cet élément que si la DLL que vous appelez attend un pointeur de données de type Huge. Si vous essayez de passer un pointeur de type Huge à une fonction qui attend un pointeur normal, l'application peut se bloquer.



Avertissement

N'essayez pas de redimensionner un tableau avec des fonctions système, telles que realloc. Vous risqueriez de bloquer le système.

Chaînes de caractères : vous pouvez spécifier le format des chaînes de caractères. Les éléments du format de chaîne de caractères sont C, Pascal, ou G.



Avertissement

Basez votre sélection du format de la chaîne de caractères sur le type de chaîne de caractères que la fonction de la bibliothèque attend. La plupart des bibliothèques standard attendent soit une chaîne de caractères C (chaîne de caractères suivie par un caractère nul) soit une chaîne de caractères Pascal (chaîne de caractères précédée par un octet de longueur). Si la fonction de la bibliothèque que vous appelez est spécifiquement écrite pour le G, vous pourriez utiliser le format Handle de chaîne de caractères, qui est un pointeur de pointeur de quatre octets d'informations sur la longueur, suivi par des données de chaîne de caractères.



Remarque

Sous Windows 3.1, vous ne pouvez pas utiliser le format Handle de chaîne de caractères ni retourner une chaîne de caractères. Sur les plates-formes qui supportent un type de retour de chaîne de caractères, la chaîne de caractères est immédiatement copiée dans un buffer. La chaîne de caractères n'est pas désallouée.



Avertissement

N'essayez pas de redimensionner un tableau avec des fonctions système, telles que realloc. Vous risqueriez de bloquer le système.

Adapter au type : vous permet de transmettre des types de données arbitraires en langage G aux DLL. Ils sont transmis de manière identique à celle du CIN. Cela signifie que :

- Les scalaires sont transmis par référence (un pointeur de scalaire est transmis à une bibliothèque).
- Les tableaux et les chaînes de caractères sont transmis sous forme de handle (pointeur de pointeur aux données). Consultez le *Interface Reference Manual*, disponible au format PDF sur les disquettes ou CD du programme.
- Les clusters sont transmis par référence.
- Les éléments scalaires des tableaux ou des clusters sont en ligne. Par exemple, un cluster qui contient un nombre est transmis comme un pointeur à une structure qui contient un nombre.
- Les clusters des tableaux sont en ligne.
- Les chaînes de caractères et les tableaux de clusters sont référencés par un handle.



Remarque

Sous Windows 3.1, vous ne pouvez pas utiliser l'option Adapter au type.

Appel de fonctions qui attendent d'autres types de données

Vous allez parfois rencontrer une fonction qui attend un type de données que le G n'utilise pas. Par exemple, vous ne pouvez pas utiliser "Appeler une fonction d'une DLL" pour transmettre un cluster ou un tableau arbitraire de données non numériques.

En fonction du type de données, vous pouvez peut-être passer les données en créant une chaîne de caractères ou d'octets qui contient une image binaire des données que vous voulez envoyer. Vous pouvez créer des données binaires en adaptant le type des éléments de données en chaînes de caractères et en les concaténant.

Vous pouvez aussi écrire une fonction de bibliothèque qui accepte les types de données utilisés par le G, ainsi que des paramètres pour construire les structures de données attendues par la fonction de bibliothèque, et qui appelle ensuite la fonction de bibliothèque.

Enfin, vous pouvez écrire une code interface node. Les code interface nodes peuvent accepter des structures de données arbitraires, mais sont difficiles à maîtriser car vous devez comprendre les formats de stockage de données du G.

Convertisseur de panneau de fonction LabWindows/CVI



Remarque

Cette fonction n'est pas disponible sur Macintosh.

Le convertisseur de panneau de fonction LabWindows/CVI automatise la procédure de conversion des drivers d'instruments écrits en LabWindows/CVI de manière à pouvoir être utilisés dans le G. Un driver d'instrument LabWindows/CVI est composé de fichiers d'en-tête et d'une source C, d'une bibliothèque de liaisons dynamiques (DLL) (**Windows**) ou d'une bibliothèque partagée (**UNIX**) contenant le code compilé et d'un fichier spécifique à LabWindows/CVI appelé fichier "panneau de fonction". Les fichiers "panneau de fonction" sont utilisés dans LabWindows/CVI de manière à ce que les utilisateurs puissent spécifier des arguments et des valeurs retournées lors de l'édition d'une instruction d'appel de fonction C en fournissant des valeurs dans une fenêtre locale.

Le convertisseur de panneau de fonction LabWindows/CVI convertit chaque fonction en un VI en utilisant les informations contenues dans le fichier CVI FP afin de déterminer le type, la représentation des données et le placement des commandes sur les faces-avant de VI. Chaque VI généré utilise un nœud "Appeler une fonction d'une DLL" sur son diagramme pour appeler la routine C appropriée dans la bibliothèque de drivers correspondante.

Bien que de nombreux drivers d'instruments soient entièrement écrits sur le diagramme sans appeler des fonctions de bibliothèque, le convertisseur de panneau de fonction LabWindows/CVI offre des avantages dans certains cas. Pour des instruments où aucun driver G n'est disponible, cet outil facilite relativement l'utilisation d'un driver CVI lorsqu'il existe. Toutefois, lorsqu'on a le choix, il vaut mieux utiliser un vrai driver G composé de diagrammes G car il est plus facile pour les clients de le visualiser et de le modifier, mais également en raison de son aspect multi-tâche qui se comporte bien au sein de l'environnement de G. Les appels de bibliothèque sont synchrones, tous les VIs exécutés en parallèle s'arrêtent pendant la durée de l'appel.



Remarque

Tous les fichiers CVI .dll nécessitent un support en exécution. Installez le moteur CVI Run-Time pour utiliser les VIs créés par le processus de conversion.

Processus de conversion

Si vous sélectionnez **Fichier>Convertir un driver d'instrument CVI .fp...**, une boîte de dialogue s'affiche et vous invite à sélectionner un fichier de panneau de fonction LabWindows/CVI. Lorsque vous sélectionnez un fichier FP, vous voyez la boîte de dialogue suivante dans laquelle vous indiquez l'endroit où enregistrer les nouveaux VIs et les fonctions de driver à convertir.



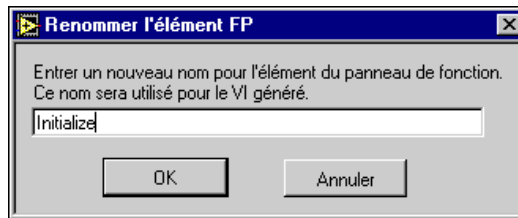
Vous spécifiez le répertoire de destination ou la bibliothèque de VIs dans la zone de texte supérieure. La destination suggérée est affichée et vous pouvez modifier le chemin afin de placer les nouveaux VIs là où vous le souhaitez. Vous pouvez utiliser le bouton **Parcourir...** pour spécifier la destination en utilisant une boîte de dialogue de fichier habituelle.

La zone de texte **Préfixe de l'instrument** affiche le préfixe de l'instrument comme le fait le fichier de panneau de fonction. Ce préfixe est attribué par CVI à tous les noms de fonctions de C dans l'arborescence des panneaux de fonction. Le convertisseur attribue aussi ce préfixe aux noms des VIs qu'il génère. Le préfixe suggéré est affiché selon la DLL, mais vous pouvez modifier ou supprimer cette chaîne de caractères.

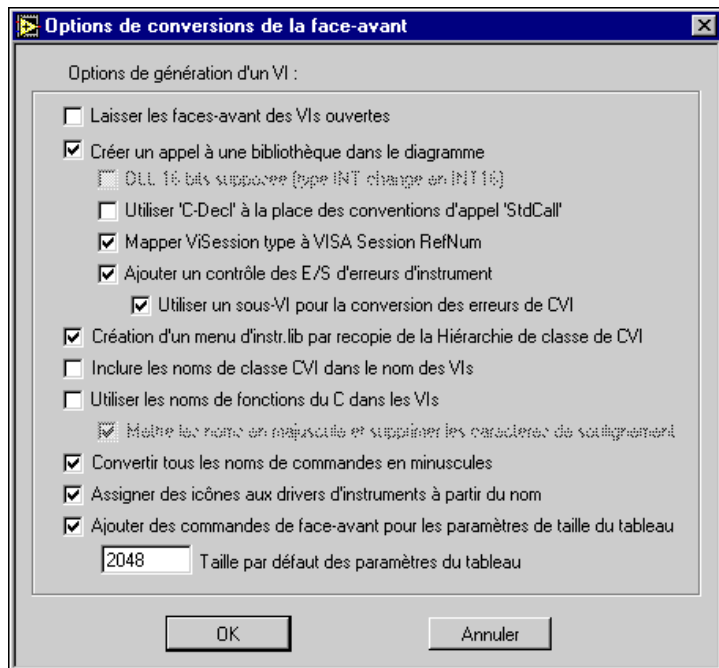
Le reste de la boîte de dialogue contient la sélection des éléments du panneau de fonction à convertir. Une boîte-liste répertorie tous les éléments de l'arborescence des panneaux de fonction, découpée de manière hiérarchique par classe, comme dans LabWindows/CVI. Les noms de

classe sont aussi répertoriés mais grisés et impossibles à sélectionner. Au départ, tous les nœuds trouvés dans le fichier "panneau de fonction" sont sélectionnés. Les éléments sélectionnés sont indiqués par une coche, ceux qui ne le sont pas n'ont pas de symbole et les noms de classe sont indiqués par un rectangle.

Si vous cliquez deux fois sur un élément, vous inversez la sélection. Les boutons **Sélectionner tout** et **Désélectionner tout** sont aussi à votre disposition. Vous pouvez également cliquer sur un élément une fois pour déplacer la barre de surbrillance sur cet élément et ensuite cliquer sur **Renommer**. Cette opération ouvre la boîte de dialogue suivante, qui vous servira à renommer un élément FP.



Le bouton **Options...** ouvre la boîte de dialogue suivante.



Les options de conversion sont les suivantes :

Laisser les faces-avant de VI ouvertes : le convertisseur laisse les VIs en mémoire avec leurs faces-avant ouvertes à la fin de la conversion plutôt que d'abandonner chaque VI une fois enregistré sur le disque (cette option est désactivée par défaut).

Créer un Appel à une bibliothèque dans le diagramme : le convertisseur place un nœud "Appeler une fonction d'une DLL" sur le diagramme de chaque VI et relie tous les terminaux de face-avant de manière appropriée. Si cette option n'est pas activée, seule la face-avant est créée ; rien n'est déposé sur le diagramme (cette option est activée par défaut).

DLL 16 bits supposé (Windows 3.1) : le convertisseur considère le type Entier du panneau de fonction de LabWindows/CVI comme INT16 plutôt que INT32 lors de la création du descripteur de type qui décrit les arguments de "Appeler une fonction d'une DLL". Cette option est nécessaire lors de l'appel de DLL créées avec des compilateurs telles que Borland C (cette option est activée par défaut).

Ajouter un contrôle des E/S d'erreurs d'instrument : le convertisseur dépose un code de gestion d'erreur sur le diagramme de chaque VI créé. Si vous sélectionnez cette option, des clusters Entrée d'erreur et Sortie d'erreur sont déposés sur la face-avant, en dessous de toutes les autres commandes et indicateurs dérivés du panneau de fonction LabWindows/CVI. Les terminaux de face-avant et "Appeler une fonction d'une DLL" sont inclus dans une structure Condition qui n'est exécutée que si le champ d'état de l'Entrée d'erreur est faux (FALSE), ce qui indique qu'il n'y a pas d'erreur (cette option est activée par défaut).

Utiliser un sous-VI pour la conversion des erreurs de CVI : dépose un sous-VI sur le diagramme de chaque VI afin de mapper les codes d'erreur du style LabWindows/CVI en clusters d'erreur du style G qui conviennent à l'alimentation du Gestionnaire général d'erreurs. La valeur entière retournée par le nœud "Appeler une fonction d'une DLL", le cluster Entrée d'erreur, et le nom du VI en cours sont transmis au sous-VI. Si une erreur est détectée par le sous-VI, elle est transmise à la Sortie d'erreur, avec un état vrai (TRUE). Si une mise en garde est détectée, elle est transmise à la Sortie d'erreur, mais avec un état faux (FALSE). Sinon, Entrée d'erreur est transmise à la Sortie d'erreur. Si cette option n'est pas activée, aucun sous-VI n'est déposé, et le diagramme est conçu de manière à ce qu'une erreur ne soit signalée dans la Sortie d'erreur que si la valeur retournée par "Appeler une fonction d'une DLL" est inférieure à zéro (cette option est activée par défaut).

Inclure les noms de classe CVI dans les noms des VIs : crée automatiquement un nom pour chaque VI selon le nom du panneau de fonction ou selon le nom de la fonction C pour chaque option de driver d'instrument. Si cette option est cochée, le nom de Classe associé au panneau de fonction LabWindows/CVI est attribué au nom de la fonction automatiquement générée.

Utiliser les noms de fonction C dans les VIs : généralement, le convertisseur conçoit les noms des VIs qu'il génère directement à partir des noms d'élément du panneau de fonction en attribuant simplement le préfixe de l'instrument et en ajoutant `.vi`. Malheureusement, cette méthode génère parfois des noms identiques aux éléments car LabWindows/CVI tolère des noms identiques pour les branches de l'arborescence du panneau de fonction. Cette option permet au convertisseur de concevoir des noms de VI à partir du véritable nom de la fonction C à laquelle les éléments FP correspondent, ce qui produit à coup sûr des résultats uniques.

Pour éviter les problèmes dus aux noms en double, le convertisseur vérifie les éléments pour déterminer s'ils sont uniques au moment de la conception de la liste des éléments de panneau de fonction à afficher. Les éléments en double sont signalés par un symbole nul et ne peuvent être sélectionnés en double-cliquant dessus. Le convertisseur affiche ensuite un bouton d'alerte lorsqu'il ouvre pour la première fois la boîte de dialogue s'il détecte de tels conflits de nom et active automatiquement l'option **Utiliser les noms de fonction C**. Si vous préférez utiliser les noms de panneau de fonction, vous pouvez désactiver cette option manuellement et renommer ensuite chaque élément qui possède un nom en conflit. Les éléments renommés conservent le nom que vous leur avez attribué même si l'option **Utiliser les noms de fonction** est ultérieurement activée.

Mettre les noms en majuscules et supprimer les caractères de soulignement : étant donné que les noms conçus à partir des noms de fonction C sont moins précis que ceux dérivés des noms d'élément de panneau de fonction, cette option tente de les améliorer en mettant en majuscules les lettres initiales et en remplaçant les caractères de soulignement par des espaces. Étant donné qu'il ne peut développer les abréviations, le résultat peut ne pas autant satisfaire que les noms FP, mais cette option est nécessaire avec les noms d'élément FP en double que vous ne voulez pas renommer individuellement.

Convertir tous les noms de commandes en minuscules : convertit les noms de commandes en minuscules pour se conformer aux standards *VXIPlug&Play*.

Assigner des icônes aux drivers d'instruments à partir du nom : affecte les icônes aux VIs en fonction du nom des VIs générés. Le convertisseur recherche les mots clés tels que initialiser, fermer, auto-test, réinitialiser, configurer ou mesurer dans le nom de la fonction et utilise l'icône correspondante. Si aucun mot clé n'est trouvé, une icône par défaut est utilisée. De plus, le préfixe de l'instrument est imprimé dans l'icône en bas à gauche, avec un maximum de sept caractères.

Taille par défaut des paramètres du tableau : lorsqu'une fonction DLL de driver d'instrument génère un tableau, la mémoire dans laquelle la fonction écrit le tableau doit être pré-allouée et transmise dans la DLL. Grâce à cette option, vous pouvez sélectionner la taille par défaut, en éléments, à allouer aux tableaux. Lorsqu'un nœud "Appeler une fonction" possède un tableau en argument, le convertisseur dépose une fonction "Initialiser un tableau" afin de créer un tableau à transmettre au nœud. La taille initiale de tels tableaux est spécifiée par l'option **Taille par défaut des paramètres du tableau**. Un avertissement est généré dans le fichier .out pour que vous puissiez facilement trouver les VIs qui contiennent cette conception et gérer les cas spéciaux de manière individuelle.

Sélectionnez **OK** dans la boîte de dialogue principale du Convertisseur de panneau de fonction LabWindows/CVI. Une boîte de dialogue de fichier s'affiche alors pour que vous puissiez sélectionner la bibliothèque correspondante au fichier FP en cours de conversion (si l'option **Créer un Appel à une bibliothèque** est activée). L'annulation de cette boîte de dialogue laisse simplement un chemin de bibliothèque non spécifié dans le nœud "Appeler une fonction d'une DLL". Elle n'annule pas la conversion.

Une fois que vous avez cliqué sur **OK**, le convertisseur affiche une boîte de dialogue d'état de fonctionnement qui montre le nom de chaque nouveau VI au fur et à mesure de sa création. Un fichier d'enregistrement appelé `prefix.out` est créé. Il répertorie tous les VIs créés et les mises en garde ou les erreurs survenues pendant la conversion. Si des mises en garde ou des erreurs se produisent, vous serez invité par le programme à contrôler ce fichier à la fin de la conversion.

Comprendre le système d'exécution du G

Ce chapitre présente le fonctionnement et l'exécution multitâche des VIs.

Grâce au système d'exécution du G, vous pouvez simultanément exécuter plusieurs VIs. De plus, vous pouvez avoir à l'intérieur d'un VI donné de nombreuses branches parallèles qui peuvent également être simultanément exécutées de manière individuelle.

Dans une utilisation normale, il n'est pas nécessaire de connaître tous les rouages du fonctionnement multitâche. Vous pouvez penser à des parties d'un diagramme qui s'exécutent en parallèle et à plusieurs VIs qui s'exécutent en parallèle. Grâce à la capacité du système d'exécution en multitâche, vous pouvez éditer ou exécuter un VI en mode Pas à pas tandis que d'autres VIs continuent leur exécution. De même, si vous vous trompez en concevant un VI dans une boucle infinie, ce dernier ne bloquera pas l'ordinateur et n'empêchera pas l'exécution des autres VIs.

Dans certaines applications où le séquençement est crucial, une meilleure compréhension du fonctionnement du système multitâche est importante. La section suivante traite de ce sujet.

Vue d'ensemble du multitâche

La plupart des ordinateurs ne possèdent qu'un seul processeur, ce qui signifie qu'une seule tâche peut être exécutée à la fois. Le fonctionnement multitâche est obtenu en exécutant une tâche pendant un court instant, puis en exécutant les autres tâches. Tant que le temps imparti à chaque tâche reste raisonnablement réduit, l'utilisateur peut avoir l'impression d'une exécution simultanée multitâche.

Les systèmes d'exploitation Windows 3.1 et Macintosh utilisent principalement une certaine forme de fonctionnement multitâche appelée multitâche coopératif. Ce dernier nécessite une programmation des logiciels qui cède la place aux autres tâches de manière périodique. La plupart des applications passent un temps non négligeable à attendre des

événements tels que l'interaction de l'utilisateur. Elles ont donc normalement un certain nombre d'opportunités pour céder la place aux autres tâches. Toutefois, si une tâche reste occupée pendant trop de temps, les autres tâches ne peuvent plus s'exécuter. Le fonctionnement multitâche coopératif ne fonctionne bien que si tous les programmes le respectent. Dans la réalité, la plupart des programmes n'observent pas si bien le fonctionnement multitâche coopératif lors de l'exécution des tâches communes telles que l'impression, le déplacement des fenêtres ou l'enregistrement des fichiers.

De nombreux systèmes d'exploitation supportent une autre forme de fonctionnement multitâche appelée multitâche préemptif. Grâce à ce dernier, le système d'exploitation gère la commutation et la planification des tâches. Un certain temps limité est imparti à chaque tâche pour qu'elle puisse s'exécuter. Une fois le temps écoulé, la tâche est obligée de s'arrêter pour qu'une autre puisse commencer son exécution. Cette commutation est gérée à des moments arbitraires sans codage particulier de la part du développeur.

Multithreading

Lorsqu'une application doit exécuter une tâche, il existe généralement plusieurs façons d'aborder le problème. La première consiste à accomplir la tâche étape par étape. Toutefois, la plupart des tâches peuvent être divisées en de plus petites tâches qui peuvent théoriquement être réalisées en parallèle. Par exemple, lorsque vous imprimez un document, l'application commence l'impression en arrière-plan et vous pouvez alors éditer d'autres documents. Le multithread est le terme qui s'applique au fonctionnement multitâche d'une seule application.

Comme avec le fonctionnement multitâche coopératif, vous pouvez concevoir des programmes qui effectuent un multithread coopératif à l'intérieur d'une application. Chaque thread doit être écrit de manière à céder de temps en temps la place pour que les autres threads ou tâches puissent s'exécuter. Etant donné que tous les threads d'une application multithread sont développés comme faisant partie du même programme, le développeur en contrôle mieux la coopération. Toutefois, il est difficile de le faire de manière uniforme tout au long d'une application et il est probable que les applications multithread coopératif perdent un certain potentiel de leur exécution parallèle.

Certains systèmes d'exploitation proposent un mécanisme pour le multithread préemptif. Le système d'exploitation gère la commutation contextuelle entre les threads du programme comme il gère les différents

programmes du système multitâche préemptif. Par conséquent, le processeur est mieux distribué aux threads actifs du programme.

Le système d'exécution du G

Le système d'exécution du G utilise le multithread préemptif sur les systèmes d'exploitation qui le supportent ainsi que le multithread coopératif. Même sur les systèmes avec un multithread préemptif, un nombre limité de threads est utilisé. Par conséquent, dans certains cas, il doit retourner à l'utilisation du multithread coopératif.

(Windows 95/NT, Solaris 2 et PowerMAX) L'application peut avoir plusieurs threads. Le système d'exécution effectue une exécution multitâche des VIs de manière préemptive à l'aide des threads. Toutefois, un nombre limité de threads est disponible. Par conséquent, pour des applications hautement parallèles, il retourne en mode multitâche coopératif lorsque le nombre de threads est épuisé. De même, le système d'exploitation supporte le multitâche de manière préemptive entre l'application et les autres tâches.

(Macintosh et Windows 3.1) L'application n'a qu'un seul thread. Le système d'exécution effectue une exécution multitâche des VIs de manière coopérative à l'aide de son propre système de répartition du temps d'exécution. L'application effectue également un multitâche coopératif avec les autres applications en cédant périodiquement un court laps de temps.

(HP-UX et Solaris 1) L'application n'a qu'un seul thread. Le système d'exécution effectue une exécution multitâche des VIs de manière coopérative à l'aide de son propre système de répartition du temps d'exécution. Le système d'exploitation supporte le multitâche de manière préemptive entre l'application et les autres tâches.

Système d'exécution de base

La description suivante concerne à la fois les systèmes d'exécution à plusieurs threads et ceux qui n'ont qu'un seul thread.

Le système d'exécution place les tâches actives dans une file d'attente. Par exemple, si vous avez trois boucles qui s'exécutent de manière parallèle, à un certain moment, une tâche s'exécute alors que les deux autres attendent dans la file d'attente. En supposant que toutes les tâches ont la même priorité, une des tâches s'exécute pendant un certain temps. Cette tâche est ensuite placée dans la file d'attente et la tâche suivante s'exécute à son tour.

Lorsqu'une tâche termine son exécution, le système la fait sortir de la file d'attente.

Le système d'exécution exécute le premier élément de la file d'attente en appelant le code généré du VI. A un certain moment, le code généré de ce VI demande au système d'exécution s'il assigne l'exécution d'une autre tâche. Dans le cas contraire, le code du VI continue son exécution.

Gestion de l'interface utilisateur dans une application à un seul thread

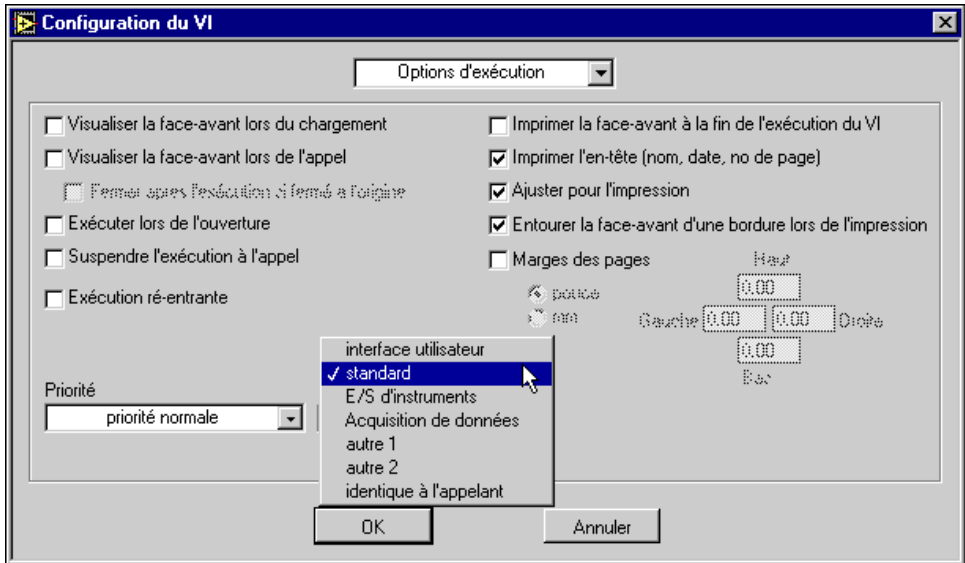
En plus de l'exécution des VIs, le système d'exécution doit coordonner l'interaction avec l'interface utilisateur. Lorsque vous cliquez sur un bouton, déplacez une fenêtre ou modifiez la valeur d'une commande curseur, le système d'exécution prend en charge la gestion de cette activité et s'assure de la poursuite de l'exécution du VI en arrière-plan.

Le système d'exécution à un seul thread assure le multitâche en commutant entre la réponse à l'interaction de l'utilisateur et l'exécution des VIs. Lorsque l'exécution retourne au VI, le système vérifie si un événement de l'interface utilisateur nécessite une certaine gestion. Dans le cas contraire, le système d'exécution retourne au VI ou s'occupe de la tâche suivante dans la file d'attente.

Lorsque vous appuyez sur des boutons ou que vous déroulez des menus, cette action peut prendre un certain temps avant de s'effectuer. Le système d'exécution continue à exécuter les VIs en arrière-plan en commutant entre la réponse à votre interaction avec la commande ou le menu et l'exécution des VIs.

Application à plusieurs threads et systèmes d'exécution multiples

Les versions à plusieurs threads ont plusieurs systèmes d'exécution. Vous pouvez affecter des VIs à un des six différents systèmes d'exécution par le biais de **Système d'exécution privilégié** dans **Configuration du VI» Options d'exécution.**



Les systèmes d'exécution parmi lesquels vous pouvez choisir sont les suivants.

- Interface utilisateur
- Standard
- E/S d'instrument
- Acquisition de données
- Autre 1
- Autre 2

L'intérêt d'avoir plusieurs systèmes d'exécution est de rendre indépendante l'exécution de deux VIs. Par défaut, les VIs s'exécutent dans le système d'exécution Standard qui s'exécute dans des threads séparés de l'interface utilisateur. Le système d'exécution E/S d'instrument permet d'empêcher les E/S de VISA, GPIB et Série d'interférer avec les autres VIs. D'une façon similaire, le système d'exécution Acquisition de données s'occupe des VIs d'acquisition de données.

Le système d'exécution Interface utilisateur se comporte de la même manière dans la version à plusieurs threads que dans celle à un seul thread. Le système Interface utilisateur prend en charge l'interface utilisateur. Les VIs peuvent s'exécuter dans le thread de l'interface utilisateur, mais le système d'exécution alterne entre le fonctionnement multitâche coopératif et la réponse aux événements de l'interface utilisateur.

Chacun des autres systèmes d'exécution possède sa propre file d'attente. Ces systèmes d'exécution ne s'occupent pas de la gestion de l'interface utilisateur. Lorsqu'un VI de ces files d'attente doit mettre à jour une commande, il transmet la responsabilité au thread Interface utilisateur.

De même, tous les systèmes d'exécution, excepté le système Interface utilisateur, possèdent deux threads responsables de l'exécution des VIs de la file d'attente. Chaque thread s'occupe de l'exécution d'une tâche. Par conséquent, si un VI appelle, par exemple, un CIN, le deuxième thread continue à exécuter les autres VIs de ce système d'exécution. Etant donné que chaque système d'exécution a un nombre limité de threads, lorsque tous les threads sont occupés, vous finissez par obtenir des tâches en attente comme dans un système à un seul thread.

Même si les VIs que vous écrivez s'exécutent correctement si vous les laissez dans le système d'exécution Standard, pensez à les affecter également à l'utilisation d'un autre système d'exécution. Si vous développez des drivers d'instruments, vous pouvez, par exemple, affecter vos VIs à l'utilisation du système d'exécution E/S d'instrument.

Même si vous laissez vos VIs en système d'exécution Standard, un avantage important du multithreading est qu'il sépare l'interface utilisateur en son propre thread. Toute activité exercée dans l'interface utilisateur (telle que le dessin sur la face-avant, la réponse aux clics de la souris, etc.) peut avoir lieu sans consommer de temps d'exécution du code du diagramme. L'affichage d'une grande quantité d'informations sur un graphe n'empêche pas l'exécution du code du diagramme. De même, l'exécution d'une longue routine de calcul n'empêche pas la réponse aux clics de la souris ou aux saisies de données du clavier de l'interface utilisateur.

Les ordinateurs qui possèdent plusieurs processeurs profitent encore plus du multithreading. Sur un système à processeur unique, le système d'exploitation anticipe les threads et partage le temps du processeur entre les threads. Sur un ordinateur à plusieurs processeurs, les threads peuvent s'exécuter simultanément sur les différents processeurs. Par conséquent, c'est véritablement plusieurs activités qui sont exercées en même temps.

Les noms E/S d'instrument, Acquisition de données, et ainsi de suite, ne sont que des suggestions pour le type de tâches que vous pouvez placer dans ces systèmes. E/S et DAQ fonctionnent dans d'autres systèmes, mais l'utilisation de ces étiquettes peut vous aider à diviser votre application et à en comprendre l'organisation.

Autre 1 et Autre 2 sont également disponibles si d'autres tâches de votre application nécessitent leur propre thread.

Nœuds synchrones/de blocage

Comme nous l'avons précédemment indiqué, certains nœuds sont synchrones, ce qui signifie qu'ils ne fonctionnent pas en multitâche avec d'autres nœuds. Dans la version à plusieurs threads, cela signifie qu'ils s'exécutent jusqu'à leur terme et le thread dans lequel ils sont exécutés est monopolisé par cette tâche tant qu'elle n'est pas terminée.

Les CIN (Code Interface Nodes), les appels de DLL et toutes les fonctions de calcul s'exécutent de manière synchrone. La plupart des VIs d'analyse et des VIs DAQ contiennent des CIN et par conséquent s'exécutent de manière synchrone. Par exemple, une FFT (Transformée de Fourier Rapide) s'exécute jusqu'à son terme sans le thread dans lequel il est exécuté et sans tenir compte du temps nécessaire à l'exécution.

Presque tous les autres nœuds sont asynchrones. Les structures, les fonctions E/S, les fonctions de séquençement et les sous-VIs que vous concevez s'exécutent tous de manière asynchrone.

Attendre, "Attendre une occurrence", les fonctions de Boîte de dialogue et les fonctions GPIB, fonctions VISA et VIs Série attendent la fin de la tâche mais sans pour autant bloquer le thread. Ces tâches sont retirées de la file d'attente jusqu'à ce que leur tâche se termine. Une fois terminée (par exemple, lorsque l'utilisateur appuie sur un bouton d'une boîte de dialogue affichée par la fonction "Boîte de dialogue"), la tâche est placée en fin de file d'attente d'exécution.

Priorité des tâches

Il existe deux méthodes pour accorder la priorité à des tâches parallèles. La première consiste à modifier le paramètre de priorité dans **Configuration du VI**. La deuxième consiste à utiliser les fonctions Attendre avec une certaine stratégie.

Dans la plupart des cas, ne modifiez pas la priorité par défaut d'un VI. L'utilisation des priorités pour contrôler l'ordre d'exécution peut ne pas produire les résultats escomptés. Si vous ne les utilisez pas correctement, les tâches avec les plus faibles priorités peuvent être complètement ignorées.

Priorité des tâches à l'aide des fonctions Attendre

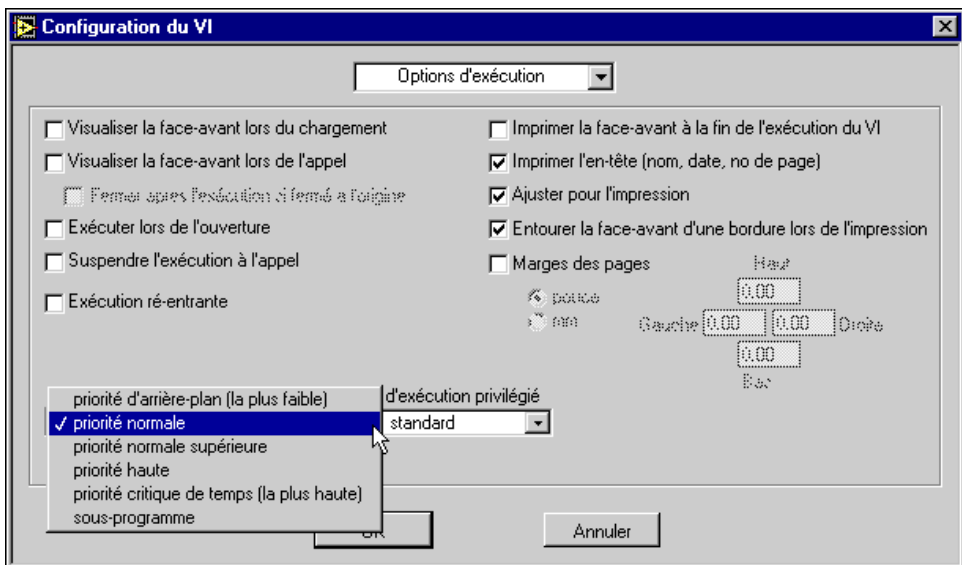
Vous pouvez utiliser la fonction Attendre pour faire en sorte que les tâches les moins importantes s'exécutent moins fréquemment. Par exemple, si plusieurs boucles sont en parallèle et que vous voulez que certaines d'entre elles s'exécutent plus fréquemment, mettez des fonctions Attendre dans les tâches qui ont la plus faible priorité. Elles accordent ainsi plus de temps aux autres tâches.

Comme il a été décrit dans la section *Nœuds synchrones/de blocage* ci-dessus, lorsqu'un diagramme attend, il est complètement retiré de la file d'attente de manière à ce que les autres tâches puissent s'exécuter.

Il est particulièrement utile d'utiliser les attentes dans les boucles qui appellent l'interface utilisateur. Une attente de 100 à 200 millisecondes ne se remarque pas forcément mais elle permet à l'application de gérer les autres tâches avec plus d'efficacité. De plus, elle libère le système d'exploitation qui peut alors se consacrer aux autres threads ou aux autres applications.

Priorité dans la configuration du VI

Vous pouvez modifier la priorité d'un VI en utilisant l'élément de menu **Priorité** dans la boîte de dialogue **Configuration du VI**.



Les six niveaux de priorité sont listés ci-dessous par ordre croissant.

- priorité d'arrière-plan (la plus faible)
- priorité normale
- priorité normale supérieure
- priorité haute
- priorité critique (la plus haute)
- priorité de type sous-programme

Les cinq premières priorités se comportent de manière identique (de la plus faible à la plus haute), tandis que la priorité de type sous-programme possède des caractéristiques supplémentaires. Ce qui suit s'applique à toutes les priorités, excepté le niveau de sous-programme.

Priorités du thread de l'interface utilisateur et du système d'exécution à un seul thread

À l'intérieur du thread de l'interface utilisateur, les niveaux de priorité sont gérés de la même manière dans un système d'exécution à un seul thread et dans un système d'exécution à plusieurs threads. Les paragraphes suivants traitent de la manière dont les priorités sont gérées dans ces configurations.

Dans un système à un seul thread comme dans le thread de l'interface utilisateur d'un système à plusieurs threads, la file d'attente du système d'exécution précédemment décrite possède plusieurs points d'entrée. Les VIs à priorité élevée sont placés dans la file d'attente devant les VIs à priorité faible. Si une tâche à priorité haute s'exécute et que la file d'attente ne contient que des tâches à priorité faible, le VI à priorité haute continue son exécution. Par exemple, si la file d'attente d'exécution contient deux VIs de chaque niveau de priorité, les VIs critiques se partagent exclusivement le temps d'exécution jusqu'à leur terme. Ensuite, c'est au tour des VIs à priorité élevée, et ainsi de suite.

L'exception se produit si les VIs à priorité élevée appellent une fonction telle que la fonction `Attendre (ms)` (consultez la section *Nœuds synchrones/ de blocage* ci-dessus pour obtenir une liste des fonctions asynchrones d'attente). Dans ce cas, les VIs à priorité élevée sont retirés de la file d'attente jusqu'à ce que l'attente ou l'E/S soit terminée, permettant aux autres tâches (probablement avec une priorité inférieure) de s'exécuter. Lorsque l'attente ou l'E/S est terminée, le système d'exécution réinsère la tâche en suspens dans la file d'attente devant les tâches à priorité faible.

De même, si un VI à priorité élevée appelle un sous-VI à priorité faible, ce sous-VI est ramené au même niveau de priorité que l'appelant pendant la durée de l'appel. Par conséquent, vous n'avez pas besoin de modifier les niveaux de priorité des sous-VIs appelés par un VI.

Utilisez les priorités avec précaution car les tâches à priorité faible peuvent facilement être privées de temps à cause des tâches à priorité élevée. Si les tâches à priorité élevée sont conçues de manière à s'exécuter pendant un long laps de temps, les tâches à priorité faible ne vont pas s'exécuter à moins que la tâche à priorité supérieure n'attende périodiquement ou n'exécute une E/S de sorte qu'elle quitte la file d'attente. Lors de l'utilisation des priorités, pensez à ajouter des attentes aux sections les plus critiques de vos diagrammes afin de libérer du temps.

Priorités dans les autres systèmes d'exécution du système à plusieurs threads

Ce chapitre a décrit précédemment les six systèmes d'exécution, qui correspondent à l'élément de menu Système d'exécution de **Configuration du VI**. En fait, pour chacune de ces six catégories, il existe un système d'exécution séparé pour chaque niveau de priorité (le niveau de priorité de type sous-programme exclus, ainsi que le système d'exécution de l'interface utilisateur). Chacun de ces systèmes d'exécution prioritaires possède sa propre file d'attente et deux threads consacrés à la gestion des diagrammes de cette file d'attente.

Plutôt que d'avoir six systèmes d'exécution, il en existe un pour le système de l'interface utilisateur, sans tenir compte de la priorité, et 25 pour les autres systèmes (cinq systèmes multipliés par 5 pour chaque niveau de priorité).

Les threads de chacun de ces systèmes d'exécution sont affectés à des niveaux de priorité du système d'exploitation en fonction de leur classification. Cela signifie qu'en exécution normale les tâches à priorité élevée obtiennent plus de temps que celles qui ont des priorités plus faibles. Comme avec les priorités du thread de l'interface utilisateur, les tâches à priorité élevée peuvent bloquer celles à priorité plus faible si leur temps d'exécution est long et sans attente.

Certains systèmes d'exploitation essaient d'éviter ce blocage en fournissant de temps en temps un meilleur niveau de priorité aux tâches à plus faible priorité. Par conséquent, sur ces systèmes, même si une tâche à priorité élevée veut s'exécuter de manière continue, les tâches à priorité plus faible peuvent quand même s'exécuter de temps en temps. Toutefois, ne comptez

pas sur cette caractéristique car elle varie d'un système à l'autre, et sur certains, les priorités des tâches et le rehaussement de priorité doivent être ajustés par l'utilisateur.

Le système de l'interface utilisateur n'a qu'un seul thread qui lui est associé. Ce thread se trouve au même niveau de priorité que celui du système d'exploitation qui est la Priorité normale pour les autres systèmes d'exécution. Par conséquent, si vous créez un VI qui s'exécute dans le système d'exécution standard avec une priorité normale supérieure, il privera l'interface utilisateur de temps. Vous obtiendrez alors une interface utilisateur lente, voire bloquée. De même, si vous créez un VI qui s'exécute avec une priorité d'arrière-plan, il s'exécutera avec une priorité inférieure à celle du thread de l'interface utilisateur.

Comme avec l'interface utilisateur, lorsqu'un VI appelle un sous-VI à priorité inférieure, le système d'exécution augmente la priorité du sous-VI au même niveau que l'appelant pendant la durée de l'appel. Par conséquent, si un VI et son sous-VI sont tous les deux affectés au même système d'exécution dans la configuration du VI, l'appel au sous-VI à priorité inférieure reste dans le même système d'exécution prioritaire. Toutefois, si le VI appelle un sous-VI à priorité supérieure, cet appel de sous-VI s'exécute dans un système d'exécution à priorité différente et supérieure.

Niveau de priorité de type sous-programme

Si vous créez un VI à priorité de type sous-programme (le plus haut niveau), le comportement change légèrement. Le niveau Priorité de type sous-programme permet à un VI de s'exécuter aussi efficacement que possible. Le compilateur compile les VIs de type sous-programme de manière à ce qu'ils ne partagent pas de temps avec les autres VIs.

Lorsqu'un VI s'exécute à une priorité de type sous-programme, il prend effectivement le contrôle du thread dans lequel il s'exécute (et il s'exécute dans le même thread que son appelant). Aucun autre VI ne peut s'exécuter dans ce thread tant que le VI à priorité de type sous-programme ne s'est pas terminé, même si un autre VI est également marqué comme sous-programme. Dans un système d'exécution à un seul thread, cela signifie qu'aucun autre VI ne s'exécute. Dans des systèmes d'exécution qui ne sont pas des interfaces utilisateur, le thread qui exécute le sous-programme ne gère pas d'autres VIs, mais le deuxième thread du système d'exécution, et les autres systèmes d'exécution, peuvent continuer à exécuter des VIs.

Non seulement le temps n'est pas partagé avec les autres VIs, mais l'exécution du VI de type sous-programme est protégée de manière à ce que les commandes et indicateurs de face-avant ne soient pas mis à jour lorsque

le sous-programme est appelé. La face-avant d'un VI de type sous-programme ne révèle rien de son exécution.

Un VI de type sous-programme peut appeler un autre VI de type sous-programme, mais ne peut pas appeler un VI d'une autre priorité. Utilisez une priorité de type sous-programme lorsque vous voulez minimiser le temps machine d'un sous-VI qui réalise des calculs simples.

De même, étant donné que les sous-programmes ne sont pas conçus dans le but d'interagir avec la file d'attente d'exécution, ils ne peuvent pas appeler de fonctions qui les retirent normalement de la file d'attente. Cela signifie qu'ils ne peuvent pas appeler de fonctions Attendre, E/S ou "Boîte de dialogue".

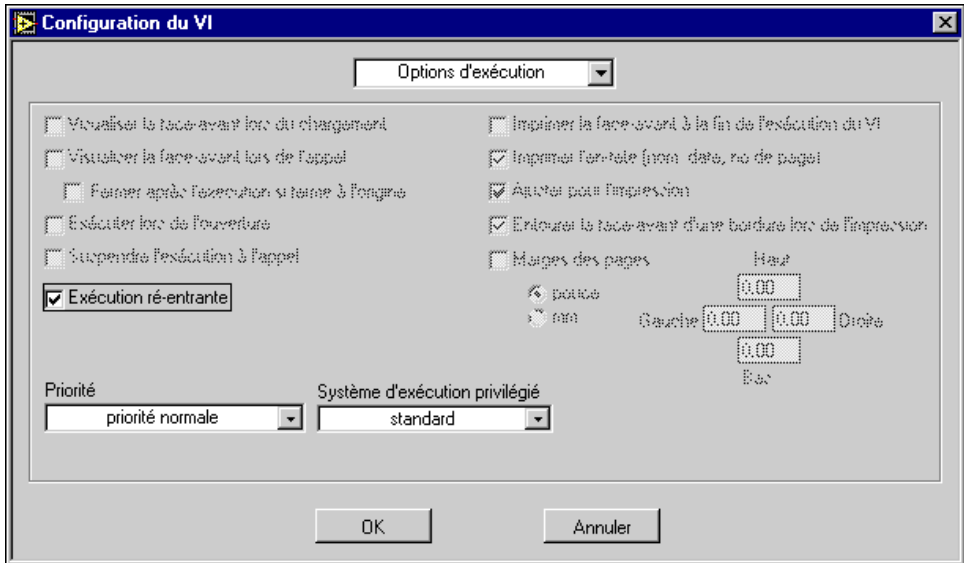
Les sous-programmes possèdent un élément supplémentaire qui peut servir dans les applications critiques. Si vous ouvrez un menu local d'un sous-VI et que vous sélectionnez "Ignorer l'appel aux sous-programmes si occupé", le système d'exécution ignore l'appel si le sous-programme exécute actuellement un autre thread. Cette opération est utile dans les boucles critiques quand les opérations exécutées par le sous-programme peuvent être ignorées sans risque et quand vous voulez éviter un délai d'attente avant que le sous-VI n'arrive à son terme. Si vous ignorez l'exécution d'un sous-VI, toutes les sorties de ce sous-VI reviennent à leur valeur par défaut pour ce type de données. Par conséquent, les sorties numériques sont mises à zéro, les sorties chaîne de caractères et tableau sont vides et les booléens sont faux (FALSE). Remarquez que c'est la valeur par défaut du type et non pas celle de l'indicateur de la face-avant du sous-VI. Si vous voulez détecter un sous-programme en cours d'exécution, faites en sorte qu'elle retourne la valeur vrai (TRUE) s'il a été exécuté avec succès. Autrement, la valeur par défaut faux (FALSE) est automatiquement retournée.

Vue d'ensemble de l'exécution ré-entrante

Normalement, le système d'exécution ne peut pas simultanément effectuer plusieurs appels au même sous-VI. Si vous essayez d'appeler un sous-VI qui n'est pas ré-entrant à partir de plusieurs locations, un seul appel s'exécute et les autres attendent leur tour pour s'exécuter. Si vous créez un VI ré-entrant (en utilisant la Configuration du VI), chaque instance de l'appel conserve son propre état d'informations. Ensuite, le système d'exécution exécute simultanément le même sous-VI à partir de plusieurs locations. L'exécution ré-entrante est utile dans les situations suivantes.

- Lorsqu'un VI attend pendant un certain temps spécifié ou jusqu'à ce qu'une limite de temps soit écoulée.
- Lorsqu'un VI contient des données qui ne sont pas partagées entre plusieurs instances du même VI, contrairement à une variable globale qui est un VI dont les données sont à partager.

Vous pouvez configurer l'exécution ré-entrante par le biais de la boîte de dialogue **Options d'exécution** de la **Configuration du VI**. Si vous sélectionnez l'option Exécution ré-entrante, plusieurs autres éléments de menu deviennent indisponibles.



Les éléments de menu suivants ne sont plus disponibles lorsque vous sélectionnez **Exécution ré-entrante**.

- Ouvrir la face-avant lors du chargement
- Visualiser la face-avant lors de l'appel
- Exécution en mode Animation
- Pas à pas

Ces éléments de menu sont désactivés car le sous-VI doit commuter entre les différentes copies des données et les différents états d'exécution pour chaque appel, ce qui rend impossible l'affichage en continu de son état actuel.

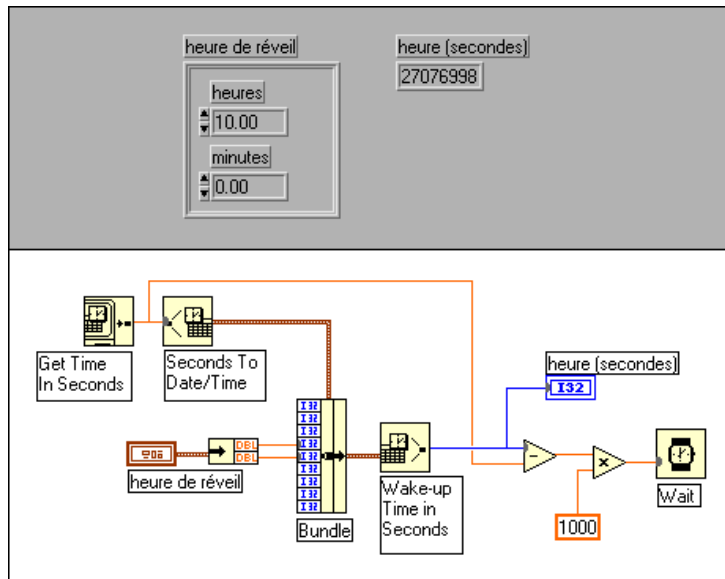
Exemples d'utilisation du mode ré-entrant

Les exemples des deux sections suivantes présentent les VIs ré-entrants.

Utilisation d'un VI d'attente



L'exemple ré-entrant suivant décrit un VI, appelé Sommeil qui accepte les heures et les minutes comme entrées et attend l'arrivée d'une heure. Si vous voulez l'utiliser simultanément à plusieurs endroits, le VI doit être ré-entrant.

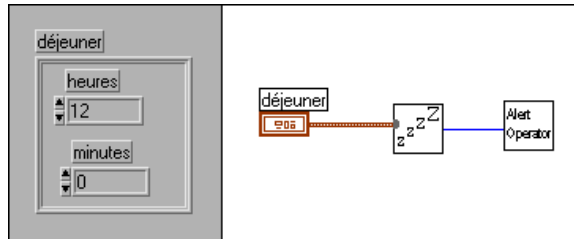


La fonction “Obtenir la date et l'heure en secondes” lit l'heure actuelle en secondes et “Secondes en date/heure” convertit cette valeur en un cluster de valeurs temporelles (année, mois, jour, heure, minute, seconde et jour de la semaine). Une fonction Assembler remplace l'heure et la minute en cours par des valeurs qui représentent une heure ultérieure du même jour dans la commande de cluster de face-avant “Heure de réveil”. L'enregistrement ajusté est alors à nouveau converti en secondes, et la différence entre l'heure en cours en secondes et l'heure future est multipliée par 1 000 afin d'obtenir des millisecondes. Le résultat est transmis à la fonction Attendre.

Midi et Pause sont deux VIs qui utilisent Sommeil comme sous-VI. Le VI Midi dont la face-avant et le diagramme sont illustrés ci-après, attend midi et s'affiche sur la face-avant pour rappeler à l'opérateur qu'il est l'heure de déjeuner. Le VI Pause ouvre un panneau pour rappeler à l'opérateur qu'il

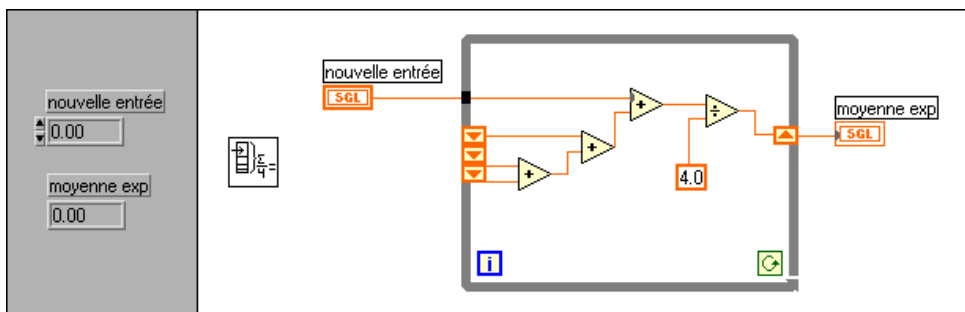
peut faire sa pause de 10 h. Le VI Pause est identique au VI Midi, aux sous-VIs locaux près.

Pour que Pause et Midi s'exécutent en parallèle, Sommeil doit être ré-entrant. Autrement, si vous lancez Midi d'abord, Pause attend que Sommeil se réveille à midi, c'est-à-dire deux heures trop tard.



Utilisation d'un VI de stockage qui n'est pas censé partager ses données

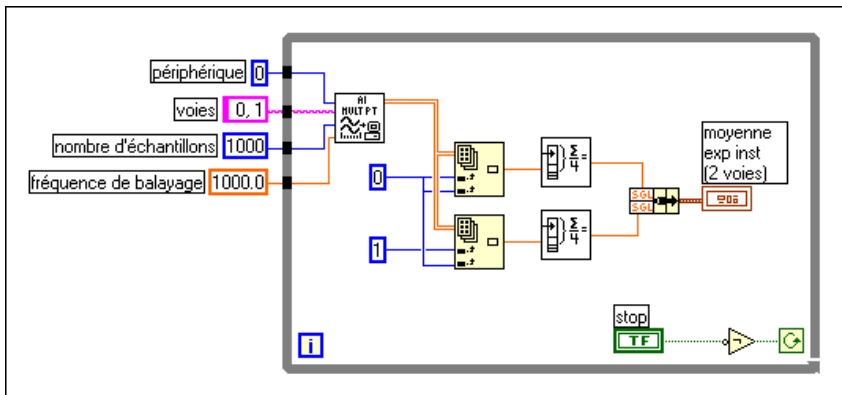
Une autre situation qui nécessite la ré-entrance est lorsque vous créez plusieurs appels à un sous-VI qui stocke des données. Supposons que vous créez un sous-VI MoyExp, qui calcule la moyenne exponentielle instantanée de quatre points de données. Afin de conserver les anciennes valeurs, MoyExp utilise un registre à décalage non initialisé avec trois terminaux de gauche. Si vous êtes un utilisateur de LabVIEW, consultez le chapitre 3, *Boucles et graphes déroulants*, de votre *Manuel de l'utilisateur LabVIEW*, pour plus d'informations sur les registres à décalage non-initialisés.



Si vous êtes un utilisateur de BridgeVIEW, consultez le chapitre 10, *Loops and Charts* du *BridgeVIEW User Manual*.

Supposons ensuite qu'un VI utilise MoyExp afin de calculer la moyenne d'exécution de deux voies d'acquisition de données. Par exemple, vous

contrôlez les tensions de deux points d'un processus et vous voulez afficher la moyenne exponentielle instantanée sur un graphe déroulant. Le diagramme contient deux nœuds MoyExp. Les appels sont alternés, un pour la Voie 0, puis un autre pour la Voie 1. Supposons que la Voie 0 s'exécute d'abord. Si MoyExp n'est pas ré-entrant, l'appel de la Voie 1 utilise la moyenne calculée par l'appel de la Voie 0, et l'appel de la Voie 1 utilise la moyenne calculée par l'appel de la Voie 1. Par contre, si MoyExp est ré-entrant, l'appel peut s'exécuter de manière indépendante sans risque de partage des données.

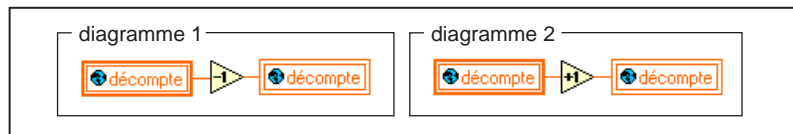


Synchronisation de l'accès aux variables globales, aux variables locales et aux ressources externes

Etant donné que le système d'exécution peut réaliser plusieurs tâches en parallèle, vous devez vous assurer que l'accès aux variables globales et locales et aux ressources se fait dans le bon ordre.

Conditions de compétition

Une *condition de compétition* se produit lorsque deux ou plusieurs morceaux de code qui s'exécutent en parallèle modifient la valeur d'une même ressource partagée qui est généralement une variable globale ou locale. Les deux diagrammes suivants sont un exemple de condition de compétition.



Le Diagramme 1 diminue de manière incrémentale la valeur du nombre et le Diagramme 2 l'augmente de manière incrémentale. Etant donné qu'il n'y a pas de dépendance des données entre les deux diagrammes, l'exécution de ces deux diagrammes peut avoir lieu dans l'ordre suivant (supposons que la valeur initiale du nombre soit 4) :

Diagramme 1 : Lecture du nombre (4)

Diagramme 2 : Lecture du nombre (4)

Diagramme 1 : Incrémentation de 4 et écriture du nombre (5 maintenant)

Diagramme 2 : Décrémentation de 4 et écriture du nombre (3 maintenant)

Bien que vous puissiez vous attendre à ce que l'exécution des diagrammes augmente le nombre de manière incrémentale et puis le diminue de manière incrémentale, le ramenant ainsi à sa valeur originale, la condition de compétition empêche une des opérations de prendre effet.

Vous pouvez remédier à cette condition de compétition de plusieurs manières. La plus simple consiste à n'avoir qu'un seul endroit dans l'application qui modifie la variable globale. Par exemple, dans le précédent exemple, tous les diagrammes qui augmentent ou diminuent de manière incrémentale le nombre de la variable globale appellent un sous-VI commun "Modifier le nombre". Ils passent un paramètre booléen pour indiquer si la variable globale doit être augmentée ou diminuée de manière incrémentale.

Etant donné qu'un VI non ré-entrant n'est exécuté qu'au nom d'un seul VI appelant à la fois, et que "Modifier le nombre" est le seul VI de l'application qui modifie le nombre de la variable globale, la condition de compétition est évitée.

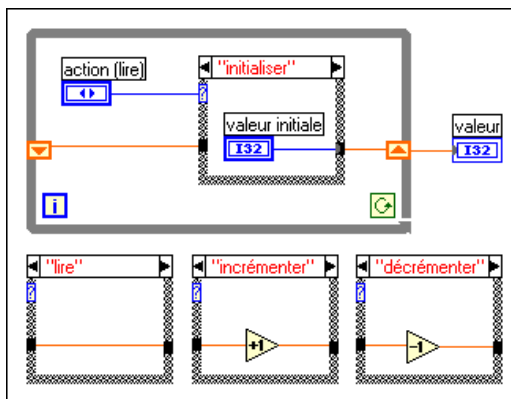


Remarque

Dans un environnement à un seul thread, il est possible d'utiliser un VI à priorité de type sous-programme pour lire, modifier et écrire une variable globale sans provoquer de condition de compétition. En effet, un VI à priorité de type sous-programme ne partage pas le thread d'exécution avec d'autres VIs. Dans un environnement à plusieurs threads, la priorité de type sous-programme ne garantit pas un accès exclusif à une variable globale car un autre VI, exécuté dans un autre thread, peut accéder à la variable globale au même moment.

Variables globales fonctionnelles

Une autre manière d'éviter les conditions de compétition associées aux variables globales consiste à ne pas utiliser de variable globale. Vous pouvez utiliser à la place des VIs qui utilisent des boucles avec des registres à décalage non initialisés pour conserver vos données de variable globale. De tels VIs sont habituellement appelés Variables globales fonctionnelles ou Variables globales de style LabVIEW 2 (les variables globales n'existaient pas dans LabVIEW 2 et seules les variables globales fonctionnelles étaient utilisées). Une variable globale fonctionnelle possède généralement un paramètre d'entrée d'action qui spécifie la fonction exécutée par le VI. Le VI utilise un registre à décalage non initialisé dans une boucle While afin de conserver le résultat de l'opération. Le diagramme suivant est une variable globale fonctionnelle qui implémente une simple variable globale de nombre. Les actions de cet exemple sont **initialiser**, **lire**, **incrémenter** et **décrémenter**.



A chaque appel du VI, le diagramme de la boucle va s'exécuter une fois. En fonction du paramètre **action**, la Condition à l'intérieur de la boucle va soit s'initialiser, soit ne pas modifier, soit augmenter ou diminuer de manière incrémentale la valeur du registre à décalage.

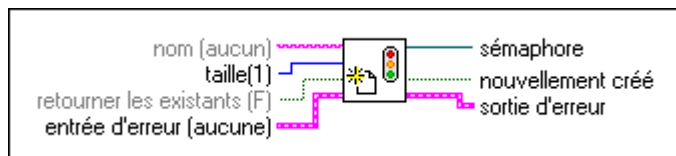
Bien que les variables globales fonctionnelles puissent être utilisées pour implémenter de simples variables globales, comme dans l'exemple précédent, elles sont particulièrement utiles lors de l'implémentation de structures de données plus complexes, telles qu'une pile ou un buffer de file d'attente. Les variables globales fonctionnelles sont également utilisées pour protéger l'accès aux ressources globales, telles que les fichiers, les instruments et les cartes DAQ, qui ne peuvent pas être représentées par une variable globale.

Sémaphores

La plupart des problèmes de synchronisation concernant les ressources globales sont facilement résolus par des variables globales fonctionnelles, puisque ce VI assure que les données qu'il contient ne sont modifiées qu'au nom d'un seul appelant à la fois. Un inconvénient de la variable globale fonctionnelle est que lorsqu'une ressource qu'elle contient doit être modifiée d'une nouvelle manière, le diagramme du VI de la variable globale doit aussi être modifié et la nouvelle action ajoutée. Dans certaines applications où l'utilisation des ressources de la variable globale change fréquemment, cet inconvénient peut être un handicap. Dans ce cas, concevez votre application de manière à ce qu'elle utilise un sémaphore afin de protéger l'accès à la ressource de la variable globale.

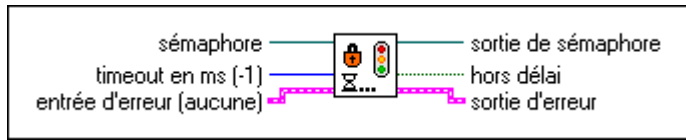
Un sémaphore, également appelé Mutex, est un objet utilisé pour protéger l'accès aux ressources partagées. Le code où les ressources partagées sont atteintes fait souvent référence à une section critique. En général, vous ne voulez exécuter qu'une seule tâche à la fois dans une section critique protégée par un sémaphore commun. Les sémaphores peuvent permettre la présence de plusieurs tâches (jusqu'à une limite prédéfinie) dans une section critique.

Vous pouvez créer un nouveau sémaphore avec un VI "Créer un sémaphore" (Create Semaphore.vi) (que vous pouvez trouver dans la palette **Fonctions»Avancé»Synchronisation»Sémaphore**). Le VI prend la valeur initiale de la taille du sémaphore comme entrée. La taille détermine le nombre de différentes tâches qui utilisent le sémaphore au même moment. A chaque fois qu'une tâche commence à utiliser le sémaphore, sa taille diminue de manière incrémentale. Lorsqu'elle atteint zéro, les tâches qui essaient d'utiliser le sémaphore doivent attendre que le sémaphore soit libéré par une autre tâche.

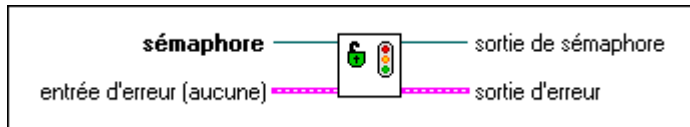


Une tâche indique qu'elle veut utiliser un sémaphore en appelant le VI "Acquérir un sémaphore" (Acquire Semaphore.vi). Lorsque la taille du sémaphore est supérieure à zéro, le VI retourne immédiatement "hors délai" = FALSE et la tâche continue. Si la taille du sémaphore est de zéro, la tâche attend que le sémaphore soit à nouveau disponible ou que le VI attende le temps spécifié. Si le VI retourne "hors délai" = TRUE, il indique que le sémaphore n'a pas été acquis et les tâches ne s'exécutent pas dans

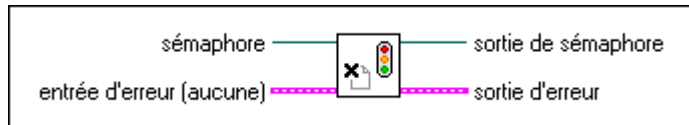
la section critique. Si le VI retourne "hors délai" = FALSE, il indique que le sémaphore a été acquis avec succès.



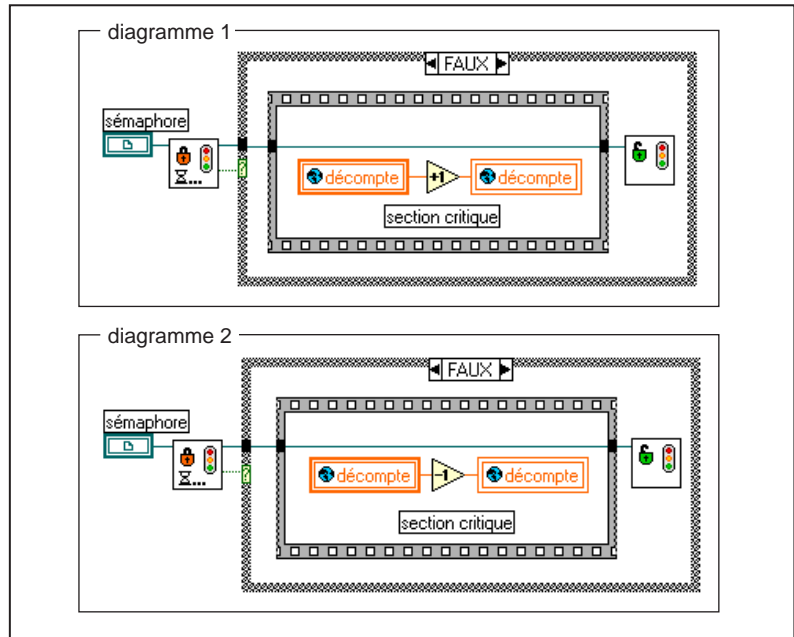
Lorsqu'une tâche acquiert avec succès un sémaphore et qu'elle termine sa section critique, elle libère le sémaphore en appelant le VI "Libérer un sémaphore" (Release Semaphore.vi). Si, au moment de la libération du sémaphore, certaines tâches attendent d'acquérir le sémaphore, la première tâche peut poursuivre son exécution. Sinon, la taille du sémaphore augmente de manière incrémentale.



Lorsqu'un sémaphore devient inutile, il est détruit en appelant le VI "Détruire un sémaphore" (Destroy Semaphore.vi). Si des VIs "Acquérir un sémaphore" attendent le sémaphore, ils retournent immédiatement "hors délai" = TRUE, ainsi qu'une erreur.



L'exemple suivant montre la manière dont un sémaphore est utilisé afin de protéger des sections critiques (augmente ou diminue de manière incrémentale une variable globale). Le sémaphore a été créé en transmettant la taille 1 au VI "Créer un sémaphore" (Create Semaphore.vi).



Chaque diagramme qui désire exécuter une section critique doit d'abord appeler un VI "Acquérir un sémaphore". Si le sémaphore est occupé (sa taille est de zéro), le VI attend que le sémaphore se libère. Lorsqu'un VI "Acquérir un sémaphore" retourne "hors délai" = FALSE, pour indiquer que le sémaphore a été acquis, le diagramme commence l'exécution de la condition FALSE. Lorsque le diagramme termine sa section critique (Séquence), le sémaphore est libéré, ce qui permet à un autre diagramme en attente de poursuivre son exécution.

Autres fonctions de synchronisation

Il existe en plus des VIs Sémaphore un certain nombre de fonctions supplémentaires et de VIs pour synchroniser des tâches parallèles. Toutes ces fonctions se trouvent dans la palette **Fonctions»Avancé» Synchronisation»Sémaphore**.

Fonctions d'occurrence : elles vous permettent de faire en sorte qu'une ou plusieurs tâches attendent qu'une autre tâche les notifie. Vous pouvez utiliser la fonction "Générer une occurrence" pour créer une occurrence que vous devez transmettre soit à une fonction "Attendre une occurrence", soit à "Définir une occurrence".

VI de notification : identiques aux occurrences, mais ils possèdent en plus une caractéristique qui vous permet de joindre un message à l'envoi de la notification. Si plusieurs VIs attendent la même notification, ils reçoivent tous le message. Et, contrairement aux occurrences, vous pouvez annuler une notification.

VI de file d'attente : comme les notifications, les VIs de file d'attente vous permettent d'envoyer des messages aux autres tâches. Toutefois, plusieurs messages peuvent être mis dans la file d'attente pour qu'au moment du retrait du message l'auditeur reçoive le plus ancien en premier. Les VIs de file d'attente sont très pratiques dans les situations où il n'y a qu'un seul "auditeur". Si un "auditeur" supprime un élément de la file d'attente, les autres ne voient pas cet élément.

Un VI peut ajouter un message à la file d'attente en utilisant "Insérer un élément de queue" (`enqueue.vi`). Pour retirer le plus vieux message de la file d'attente, vous pouvez utiliser "Supprimer un élément de queue" (`dequeue.vi`). Lorsque vous créez une file d'attente en utilisant Ouvrir la référence de la file d'attente, vous devez spécifier si elle a une taille limitée. Si vous créez une file d'attente limitée et que vous essayez d'ajouter des données à la file d'attente alors que la limite est atteinte, l'opération "Insérer un élément de queue" (`enqueue.vi`) attend qu'un auditeur retire des données en utilisant "Supprimer un élément de queue" (`dequeue.vi`) avant d'ajouter un élément supplémentaire.

VI de rendez-vous : vous pouvez utiliser les VIs de rendez-vous lorsque plusieurs tâches parallèles doivent être synchronisées à un même moment. Chaque tâche qui atteint le rendez-vous attend le nombre de tâches d'attente spécifié et, à ce moment, toutes les tâches continuent leur exécution.

Conseils d'utilisation des systèmes d'exécution et des priorités

Voici quelques conseils d'utilisation des options des systèmes d'exécution décrits dans ce chapitre.

Il est important de comprendre que même si certaines semblent compliquées, la plupart des applications ne nécessitent pas l'utilisation des niveaux de priorité ou des systèmes d'exécution alternés. Le système d'exécution prend automatiquement en charge le fonctionnement multitâche de vos VIs.

Par défaut, tous les VIs sont créés de manière à être exécutés dans un système d'exécution standard à une priorité normale. Dans un système à plusieurs threads, l'activité de l'interface utilisateur est gérée par un thread

séparé, pour que vos VIs soient isolés de l'interaction de l'interface utilisateur. Même si vous utilisez un système à un seul thread, celui-ci essaie d'intercaler la gestion de l'interaction de l'interface utilisateur dans l'exécution de vos VIs, ce qui donne des résultats similaires.

En général, la manière la plus facile et la plus sûre d'accorder des priorités à l'exécution consiste à utiliser les fonctions Attendre afin de ralentir les boucles à plus faible priorité de votre application. Ceci est particulièrement utile dans des boucles des VIs de l'interface utilisateur, puisque des délais de 100 à 200 millisecondes sont imperceptibles aux yeux des utilisateurs.

Si vous utilisez des priorités, faites-le avec précaution. Si vous concevez des VIs à priorité élevée qui s'exécutent pendant un long laps de temps, pensez à ajouter des attentes aux VIs des sections les moins critiques pour qu'ils partagent le temps avec les tâches à priorité plus faible.

Manipulez les variables globales, locales et les autres ressources externes avec précaution car elles peuvent être manipulées par d'autres tâches. Utilisez une des techniques de synchronisation décrites dans ce chapitre afin de protéger l'accès à ces ressources.

La manière dont l'utilisation de la priorité est conçue dans les systèmes à un seul thread et à plusieurs threads fournit pratiquement les mêmes résultats avec des VIs identiques sur les deux systèmes. Toutefois, il y aura de légères différences de séquençement. Si vous distribuez vos VIs à des clients qui utilisent un système d'exploitation différent, nous recommandons l'essai de vos applications sous leurs conditions. Si vous utilisez le système à plusieurs threads, vous pouvez faire en sorte qu'il se comporte comme une version à un seul thread en désactivant la préférence **Exécuter en utilisant plusieurs threads** dans la section **Performances** de la boîte de dialogue **Préférences**.

Gestion de vos applications

Ce chapitre décrit comment gérer les fichiers dans votre application en G.

Arrangement des fichiers à l'aide des bibliothèques de VIs

Vous pouvez grouper plusieurs fichiers de VIs en les enregistrant ensemble sous forme de bibliothèque de VIs (par convention, le nom de ces fichiers se termine par l'extension `.lib`). Ou vous pouvez enregistrer des VIs en tant que fichiers individuels et les grouper au sein de répertoires. Pour une comparaison des deux méthodes, consultez la section *Enregistrement des VIs* du chapitre 2, *Edition des VIs*.

Si vous utilisez des bibliothèques de VIs, vous pouvez diviser votre application en plusieurs bibliothèques de VIs. Mettez les VIs principaux dans une bibliothèque de VIs, et créez d'autres bibliothèques de VIs contenant des VIs séparés par fonction. Il est plus facile de gérer vos VIs si vous organisez vos fichiers de cette manière.

Une autre raison pour laquelle il est avantageux de répartir vos VIs entre plusieurs bibliothèques de VIs est que plus la taille de votre bibliothèque de VIs augmente, plus il faut de temps pour enregistrer les fichiers. Lorsque vous enregistrez un VI dans une bibliothèque de VIs, la bibliothèque est copiée dans le répertoire temporaire, le VI est compressé et enregistré dans la bibliothèque de VIs, et la bibliothèque de VIs est recopiée dans son répertoire d'origine. Le processus de copie permet d'éviter une corruption accidentelle du fichier original, mais prolonge également le temps nécessaire à l'enregistrement des VIs.

Une bonne règle générale est d'éviter de créer des bibliothèques de VIs d'une taille supérieure à 1 Mo. Remarquez qu'il n'y a pas véritablement de limite sur la taille d'une bibliothèque et que le temps que mettent les VIs à se charger à partir des bibliothèques de VIs reste sensiblement le même quelle que soit la taille de la bibliothèque de VIs. Cependant, le temps nécessaire à l'enregistrement de VIs dans une bibliothèque de VIs augmente à mesure qu'augmente la taille de cette bibliothèque.

Une autre façon de réduire le temps nécessaire à l'enregistrement des VIs, qu'ils soient ou non dans des bibliothèques de VIs, est de vous assurer que votre répertoire temporaire se trouve sur le même lecteur (sur la même partition, sous UNIX) que vos VIs. Vous pouvez copier des fichiers plus rapidement si la source et la destination sont sur le même lecteur (ou la même partition).

Si vous utilisez des bibliothèques de VIs, vous pouvez repérer les VIs principaux en utilisant **Fichier»Editer une bibliothèque de VIs...** à partir de l'option de menu pour affecter à certains VIs le statut de VI principal. Le G liste ces fichiers au début de la liste de fichiers dans la boîte de dialogue de fichier. Il n'y a pas d'option similaire pour les répertoires, mais vous pouvez repérer vos VIs principaux en plaçant des sous-VIs dans des sous-répertoires.

Sauvegarde de vos fichiers

Les accidents sont des choses qui arrivent. Les disques durs tombent en panne et les systèmes de fichiers sont altérés. Il est préférable de faire souvent des copies de sauvegarde de vos VIs, peut-être même une fois par jour. Copiez-les sur un autre lecteur ou sur une bande magnétique.

A titre d'exemple des problèmes qui peuvent survenir, un utilisateur a accidentellement corrompu la bibliothèque de VIs qui contenait plusieurs semaines de travail. Lorsqu'il lui a été demandé d'indiquer une destination pour un fichier de données, cet utilisateur a sélectionné sa bibliothèque de VIs, écrivant ainsi des données sur sa propre bibliothèque de VIs. Heureusement, il a été capable de récupérer la plupart du contenu de sa bibliothèque de VIs, mais seulement parce qu'il avait conservé des copies de son travail.

Distribution des VIs

Lorsque vous êtes prêt à distribuer vos VIs à d'autres machines ou à d'autres utilisateurs, vous devez établir si vous voulez distribuer le code source de diagramme éditable ou si vous voulez masquer ou retirer le diagramme. Les VIs peuvent être enregistrés sans leurs diagrammes. Ceci réduit la taille du fichier sur le disque et évite que quelqu'un ne change accidentellement le code source, mais cela empêche également les utilisateurs de transférer le VI vers une autre plate-forme ou de mettre à niveau le VI pour une version future de l'environnement de développement. Si ces aspects sont importants pour vous, vous pouvez envisager de protéger vos diagrammes au moyen d'un mot de passe. Le

code source reste disponible, mais ne peut être visualisé et modifié que si vous entrez un mot de passe.

A l'aide de la boîte de dialogue **Enregistrer avec options**, vous pouvez retirer des diagrammes de vos VIs. Si vous enregistrez vos VIs sans un diagramme, le VI ne peut pas être modifié par d'autres utilisateurs. Si vous enregistrez vos VIs sans diagramme, faites attention de ne pas écraser vos versions originales.



Remarque

Les VIs sans diagramme ne peuvent pas être convertis pour des versions futures de l'application, ni pour d'autres plates-formes. Lors d'une conversion, vous devez recompiler vos VIs. Sans diagramme, un VI ne peut pas être recompilé.

Prenez également en compte l'environnement dans lequel les utilisateurs exécutent vos VIs. Voulez-vous que les utilisateurs disposent d'une version de développement ou d'une application exécutable ?

Une application exécutable est appropriée lorsque vous ne voulez pas que vos utilisateurs apportent des modifications à vos VIs. Une application exécutable contient des menus simplifiés, et les utilisateurs ne peuvent ni éditer les VIs ni visualiser les diagrammes.

Les applications exécutables sont construites à l'aide des bibliothèques Application Builder de LabVIEW. Lorsque vous construisez une application exécutable avec ces bibliothèques, vous devez répondre à ces trois questions.

- Est-ce que je veux incorporer une bibliothèque de VIs dans l'application ?
- Est-ce que je veux que l'application dispose d'un élément de menu **Ouvrir** ?
- Est-ce que je veux que l'application dispose d'un élément de menu **Quitter** ?

Si vous choisissez d'incorporer une bibliothèque de VIs dans l'application, la bibliothèque est combinée avec un moteur d'exécution en un fichier unique. Lorsque ce fichier est lancé, il ouvre automatiquement tous les VIs principaux de la bibliothèque. Si vous n'incorporez pas de bibliothèque de VIs, l'application peut ouvrir n'importe quel VI lorsqu'elle est lancée, à condition que le VI soit enregistré avec un système de développement pour cette plate-forme.

Si vous activez l'élément de menu **Ouvrir**, vous pouvez utiliser l'application pour ouvrir et exécuter n'importe quel VI dans le système de

fichiers, qu'une bibliothèque de VIs soit incorporée dans l'application ou non. En incorporant une bibliothèque de VIs, vous pouvez créer une application complètement autonome qui empêche l'utilisateur, ou le client, d'accéder aux VIs sources, même si l'utilisateur a la version de développement.

Si vous envisagez d'expédier plusieurs séries de VIs au même client, une seule application exécutable munie de l'élément de menu **Ouvrir** s'avère plus efficace que des applications incorporées séparées. Ceci est vrai parce que chaque application incorporée contient une copie du code exécutable, qui est d'environ 2 à 3 Mo.

Pour créer une application exécutable, vous devez avoir les bibliothèques Application Builder, qui sont vendues séparément. Si vous n'êtes pas un utilisateur de LabVIEW, consultez les *Notes d'informations de l'Application Builder de LabVIEW* jointes au logiciel Application Builder pour plus de détails sur la manière de construire une application.



Remarque

Etant donné que le système d'exécution BridgeVIEW est constitué de plusieurs pièces séparées, l'Application Builder ne convient pas pour construire des applications exécutables BridgeVIEW.

VIs protégés par un mot de passe

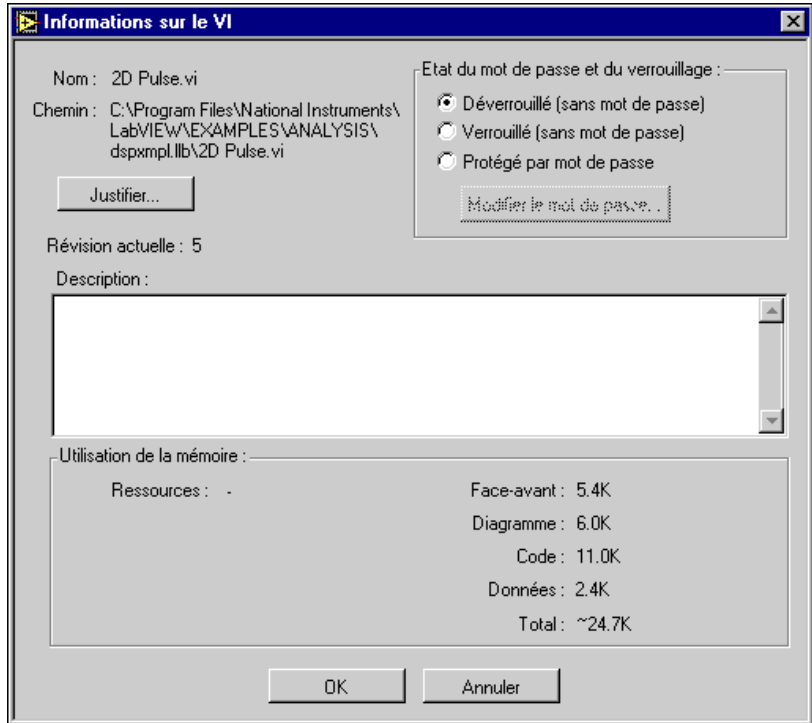
Lorsque vous distribuez des VIs à d'autres personnes, vous pouvez souhaiter interdire aux autres de visualiser le code source que vous avez créé dans le diagramme, et vous pouvez souhaiter interdire aux autres d'éditer le VI de quelque manière que ce soit.

Vous pouvez enregistrer les VIs sans les diagrammes, mais cela empêche le logiciel de recompiler les VIs lorsque vous les importez sur une autre plateforme ou une nouvelle version de logiciel en langage G.

Pour interdire aux autres utilisateurs de visualiser les diagrammes et d'éditer les VIs, vous pouvez protéger les VIs avec des mots de passe. Afin de visualiser le diagramme d'un VI protégé par un mot de passe, ou pour éditer ce VI de quelque manière que ce soit, l'utilisateur doit entrer le mot de passe correct du VI.

Pour protéger un VI par un mot de passe, sélectionnez **Fenêtres» Infos sur le VI...** ou sélectionnez **Fichier»Enregistrer avec options**.

Au moyen de la boîte de dialogue **Informations sur le VI**, représentée dans l'illustration suivante, vous pouvez ajouter une protection par mot de passe à un VI unique.



Ci-dessous figurent les descriptions des trois options disponibles dans la boîte de dialogue **Informations sur le VI** pour protéger un VI.

Déverrouillé (sans mot de passe) : indique que le VI peut être édité et qu’il n’est pas protégé par un mot de passe. Si vous sélectionnez “Verrouillé (sans mot de passe)”, le VI passe immédiatement dans l’état déverrouillé. Si l’état précédent était “Protégé par mot de passe”, et si le mot de passe pour ce VI n’est pas déjà en mémoire, une boîte de dialogue vous invite à entrer le mot de passe.

Verrouillé (sans mot de passe) : le VI est verrouillé, si bien que vous ne pouvez pas l’éditer, mais vous pouvez l’exécuter et visualiser son diagramme. Si vous passez de “Déverrouillé (sans mot de passe)” à cette option, la transition s’effectue immédiatement. Cependant, si l’état antérieur était “Protégé par mot de passe”, vous devez entrer le mot de passe s’il n’est pas en mémoire.

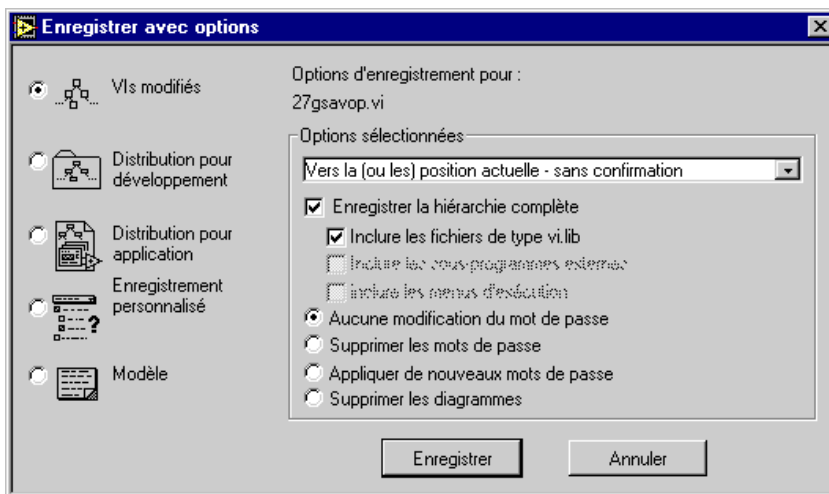
Protégé par mot de passe : cet état indique que le VI est protégé par un mot de passe. Les VIs protégés par un mot de passe sont semblables aux VIs verrouillés, mais vous ne pouvez pas accéder au diagramme sans entrer

un mot de passe. Lorsque vous voulez passer de cet état à un autre, une boîte de dialogue vous demande le mot de passe, si celui-ci n'est pas déjà en mémoire.

Pour plus d'informations sur les mots de passe, consultez la section suivante, *Boîte de dialogue "Enregistrer avec options"*.

Boîte de dialogue "Enregistrer avec options"

La boîte de dialogue **Enregistrer avec options** vous permet d'enregistrer facilement une hiérarchie entière de VIs en vue d'une distribution. Vous pouvez utiliser cette boîte de dialogue pour enregistrer de manière sélective une hiérarchie entière, sans les VIs figurant dans `vi.lib`, et enregistrer tous les sous-programmes externes appelés par vos VIs dans votre hiérarchie. La boîte de dialogue **Enregistrer avec options** est représentée dans l'illustration suivante.



En choisissant parmi les options qui se trouvent à gauche de la boîte de dialogue **Enregistrer avec options**, vous pouvez sélectionner l'une des options d'enregistrement prédéfinies. Lorsque vous sélectionnez des options, la zone située à droite illustre le comportement de votre sélection actuelle. Vous pouvez personnaliser ce comportement en sélectionnant des options spécifiques dans la section de droite de la boîte de dialogue.

- **VIs modifiés** : enregistre toutes les modifications apportées au VI le plus en avant et à ses sous-VIs. Cette option est utile pour enregistrer rapidement des modifications.

- **Distribution pour développement** : enregistre tous les VIs, commandes et sous-programmes externes ne figurant pas dans `vi.lib` dans un endroit unique (soit un répertoire, soit une bibliothèque). Cette option offre un moyen facile d'enregistrer une hiérarchie qui doit être portée vers un autre système de développement en G, étant donné que l'autre système de développement a sa propre `vi.lib`.
- **Distribution pour application** : enregistre tous les VIs, commandes et sous-programmes externes, y compris ceux qui figurent dans `vi.lib`, dans un endroit unique et supprime les diagrammes de tous les VIs. Si vous voulez créer une application exécutable avec une bibliothèque incorporée, vous pouvez utiliser cette option. Pour créer effectivement une application exécutable, vous devez avoir les bibliothèques Application Builder, qui sont vendues séparément.
- **Enregistrement personnalisé** : sélectionnez les options spécifiques que vous voulez dans la zone **Options sélectionnées** de la boîte de dialogue, si aucune des options prédéfinies ne répond à vos besoins.
- **Modèle** : enregistre le VI sous forme de modèle de VI, avec l'extension de fichier `.vit`. Lorsqu'un modèle de VI est chargé, il crée une copie de lui-même qui doit être enregistrée sous un nouveau nom.

Le bouton **Changer le mot de passe...** de la boîte de dialogue **Infos sur le VI** n'est pas disponible si le VI n'est pas protégé par un mot de passe. Lorsque vous décidez de changer un mot de passe, une boîte de dialogue vous demande le mot de passe, si celui-ci n'est pas déjà en mémoire.

Au moyen de la boîte de dialogue **Enregistrer avec options**, représentée dans l'illustration précédente, vous pouvez ajouter une protection par mot de passe à un VI ou à plusieurs VIs. Les descriptions suivantes expliquent comment le verrouillage et la protection par mot de passe sont affectés par les options de la boîte de dialogue **Enregistrer avec options**.

- **Aucune modification du mot de passe** : si vous sélectionnez cette option, cela ne donne lieu à aucune modification.
- **Supprimer les mots de passe** : si vous sélectionnez cette option, tous les VIs que vous n'avez pas enregistrés sont enregistrés sans protection par mot de passe. Si vous rencontrez des VIs protégés par mot de passe pendant l'enregistrement, pour chaque mot de passe (pas nécessairement pour chaque VI) associé à un VI ou à un groupe de VIs, une boîte de dialogue vous demande le mot de passe de chacun d'eux. Pour supprimer les mots de passe, vous devez connaître le mot de passe pour chaque VI ou groupe de VIs.

- **Appliquer de nouveaux mots de passe** : en sélectionnant cette option, vous protégez les VIs par un mot de passe. Si, pendant l'enregistrement, vous rencontrez des VIs qui sont déjà protégés par un mot de passe, pour chaque mot de passe (pas nécessairement pour chaque VI) associé à un VI ou un groupe de VIs, une boîte de dialogue vous demande le mot de passe de chacun d'eux. Pour appliquer de nouveaux mots de passe à des VIs ou à des groupes de VIs, vous devez connaître le mot de passe pour chaque VI ou groupe de VIs.
- **Supprimer les diagrammes** : cette option enregistre vos VIs sans diagramme, afin que les VIs soient moins encombrants. Les VIs occupent moins de place sur votre disque dur, mais vous ne pouvez pas les adapter à une autre version de votre logiciel, ni à d'autres plates-formes.

Concevoir des applications avec plusieurs développeurs

Si plusieurs développeurs travaillent sur le même projet, consacrez un peu de temps à définir les responsabilités de conception pour garantir le bon fonctionnement du projet. Vous pouvez aussi envisager d'utiliser le Professional G Developers Toolkit, qui fournit des outils de contrôle du code source apportant une aide précieuse dans les environnements où collaborent plusieurs développeurs.

Commencez par la conception du niveau principal de votre application et tracez-en les grandes lignes. Créez des *VIs à souche* pour les principaux composants de votre application. Un VI à souche est un prototype de sous-VI. Il a des entrées et des sorties, mais il est incomplet. Il sert à réserver un emplacement pour un développement ultérieur du VI, et il vous permettra alors d'ajouter une fonctionnalité. La création de VIs à souche, pour les utilisateurs de LabVIEW, est décrite dans le chapitre 28, *Conception de programmes*, du *Manuel de l'utilisateur LabVIEW*. Les utilisateurs de BridgeVIEW peuvent consulter le chapitre 15, *Conception de programmes*, du *BridgeVIEW User Manual*.

Une fois que vous avez vérifié que les VIs à souche apportent la fonctionnalité que vous souhaitez, vous pouvez déterminer sur quels composants doivent travailler individuellement les développeurs. Pour simplifier les choses, vous pouvez placer les principaux VIs à souche dans des répertoires séparés ou dans des bibliothèques de VIs qui peuvent alors être gérés par différents développeurs.

Conserver des copies maîtresses

Il est préférable de conserver les copies *maîtresses* des VIs de votre projet sur un seul ordinateur. Vous pouvez instituer une politique d'entrée/sortie permettant de contrôler les modifications apportées à vos VIs. Selon cette politique, un développeur peut emprunter une copie des VIs, de la même façon que l'on emprunte un livre dans une bibliothèque. Pendant ce temps, les autres ne touchent pas aux fichiers que le développeur a empruntés. Une fois qu'il a apporté les changements souhaités, le développeur renvoie les VIs finis.

Le Professional G Developers Toolkit contient des outils de contrôle de code source qui peuvent rendre relativement faciles l'utilisation partagée des fichiers et cette politique d'entrée/sortie. Il contient également un utilitaire que vous pouvez utiliser pour comparer les VIs et visualiser leurs différences. Ceci peut être extrêmement utile pour vérifier exactement quelles modifications ont été apportées entre différentes versions d'un VI.

Fenêtre Historique du VI

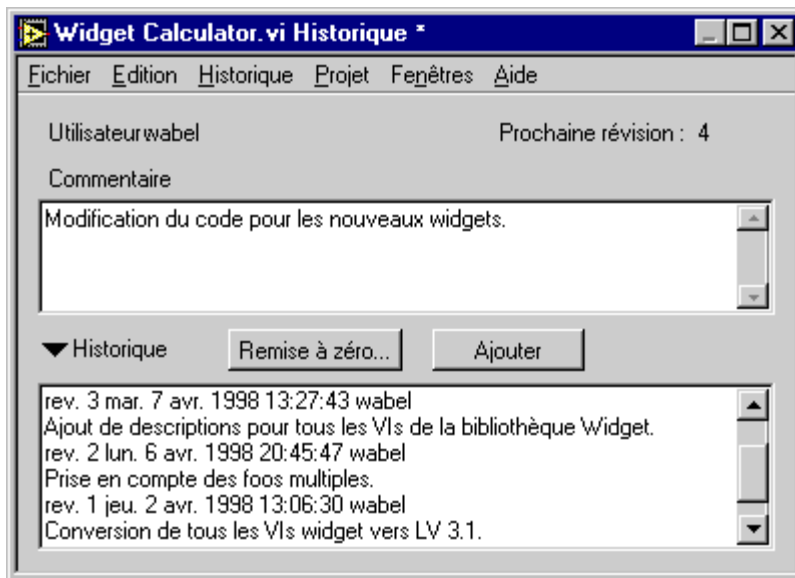
En plus de sa face-avant et de son diagramme, un VI a une fenêtre d'historique qui affiche l'historique du développement du VI, y compris les numéros de révision. Chaque utilisateur qui modifie un VI peut noter les modifications effectuées dans l'historique. Cette fonction aide les utilisateurs à garder un registre des modifications apportées à un VI au fur et à mesure qu'elles sont effectuées. Cependant, l'historique du VI n'offre pas de méthode de comparaison de deux VIs pour détecter les différences (consultez la description du Professional G Developers Toolkit pour obtenir plus d'informations sur un utilitaire qui permet de remplir cette fonction). Le G enregistre l'historique dans le cadre du VI. Il consiste en une série de commentaires entrés par les utilisateurs qui apportent des modifications au VI.



Remarque

Tant que vous ne sélectionnez pas l'option Enregistrer les commentaires générés par LabVIEW dans la boîte de dialogue Configuration du VI» Documentation ou Edition»Préférences»Historique, les commentaires ne sont pas automatiquement inscrits dans l'historique d'un VI. Toute personne qui apporte des modifications au VI doit taper les informations et garder l'historique à jour.

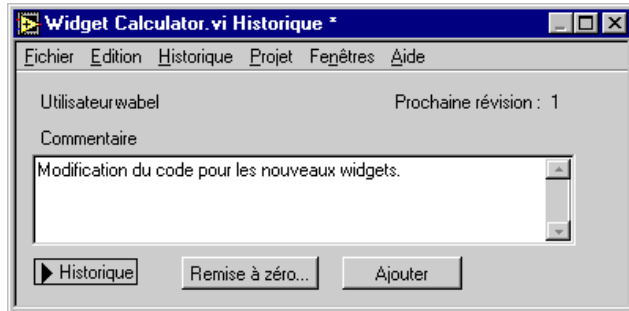
La fenêtre Historique est disponible lorsque vous éditez le VI en sélectionnant **Fenêtres»Historique** ou en utilisant l'équivalent clavier <Ctrl-y> (**Windows**), <commande-y> (**Macintosh**), <meta-y> (**Sun**) ou <Alt-y> (**HP-UX**). L'illustration suivante montre un exemple de fenêtre Historique.



Lorsque vous éditez le VI, vous pouvez taper une description des modifications importantes dans la boîte de commentaires qui se trouve près du haut de la fenêtre. Une fois que vous avez terminé d'apporter des modifications, ajoutez le commentaire à l'historique en cliquant sur le bouton **Ajouter**. S'il y a un commentaire dans la boîte de commentaires au moment où vous enregistrez le VI, il est ajouté automatiquement à l'historique. Tout commentaire que vous ajoutez devient une partie permanente de l'historique. Vous ne pouvez pas modifier le contenu de l'historique, aussi relisez vos commentaires avant de les ajouter.

La boîte en bas de la fenêtre affiche l'historique du VI. La boîte de dialogue montre un en-tête pour chaque commentaire de l'historique qui comprend le numéro de révision du VI, la date et l'heure, ainsi que le nom de la personne qui a effectué la modification. Vous pouvez voir trois commentaires ajoutés à cet historique, et chacun d'eux est affiché avec un en-tête. Etant donné que l'illustration précédente n'est qu'un exemple, les commentaires sont courts ; vous pouvez leur donner la longueur que vous voulez lorsque vous éditez vos propres VIs.

La fenêtre Historique est relativement grande lorsque l'historique est affiché, mais vous pouvez la réduire en cliquant sur la flèche **Historique** (le petit triangle noir à gauche du mot). Ceci rend la fenêtre suffisamment petite pour qu'elle ne vous gêne pas pendant que vous éditez le VI. Lorsque vous voulez voir à nouveau l'historique, cliquez sur la même flèche et l'historique réapparaît. Un exemple est représenté dans l'illustration suivante.



Vous pouvez également changer la taille des boîtes d'historique et de commentaires en redimensionnant la fenêtre.

Numéros de révision

Le numéro de révision fait aussi partie de l'historique du VI et constitue une manière facile de voir si le VI a été modifié (et comment il a été modifié, si vous avez noté des commentaires expliquant vos modifications). Le numéro de révision commence à zéro et augmente d'une unité à chaque fois que vous enregistrez le VI. Le numéro de révision actuel (celui qui figure sur le disque) apparaît dans la boîte de dialogue **Obtenir des infos...** Il est également affiché dans la barre de titre du VI si l'option correspondante est cochée dans la boîte de dialogue **Edition»Préférences»Historique**.

Le numéro affiché dans la fenêtre Historique ou dans la fenêtre du VI est le numéro de révision suivant ; c'est-à-dire, le numéro qui sera enregistré sur le disque si vous enregistrez le VI. C'est le numéro de révision actuel plus un. Lorsque vous ajoutez un commentaire à l'historique, le numéro de révision suivant est inclus dans l'en-tête du commentaire. Par exemple, si vous cliquez sur le bouton **Ajouter** dans l'exemple, votre commentaire s'ajoute à l'historique avec le numéro de révision 4. Si vous enregistrez le VI, le numéro de révision actuel devient 4, et le commentaire qui

s'applique aux modifications que vous venez d'effectuer est accompagné du même numéro.



Remarque

Le numéro de révision n'augmente pas d'une unité lorsque vous enregistrez le VI si les seules modifications que vous avez apportées sont des modifications de l'historique du VI.

Dans de nombreux cas, il manque des numéros de révision entre les commentaires, parce que vous avez enregistré le VI sans entrer de commentaire. Ceci n'est pas un problème. En fait, le numéro de révision est utile parce qu'il est indépendant. Supposez que vous receviez une copie d'un VI provenant d'un autre utilisateur. Plus tard, vous voulez savoir si cet autre utilisateur a mis à jour le VI depuis que vous avez reçu votre copie. Vous pouvez facilement le savoir en regardant les numéros de révision, même si l'autre utilisateur a apporté une modification sans ajouter de commentaire.

Si le numéro de révision change uniquement lorsque vous ajoutez un commentaire, c'est un numéro de commentaire, pas un numéro de révision. Le numéro de révision est encore plus efficace lorsqu'il est combiné à l'historique. Si les utilisateurs ajoutent des commentaires à l'historique, vous pouvez savoir quelles modifications ont été apportées depuis que vous avez reçu une copie.

Réinitialisation des informations de l'historique

Sous la boîte de commentaires se trouve un bouton intitulé **Remise à zéro**. Le fait d'appuyer sur **Remise à zéro** efface l'historique et, en option, remet à zéro le numéro de révision. Ceci est utile lorsque vous copiez un VI et que vous voulez supprimer toutes les informations antérieures pour le nouveau VI (pas d'historique).

Etant donné que l'historique est strictement un outil de développeur, l'historique est automatiquement supprimé lorsque vous supprimez le diagramme d'un VI. La fenêtre Historique n'est pas disponible dans la version exécutable, mais le numéro de révision est disponible dans la boîte de dialogue **Informations sur le VI**, même pour les VIs dont les diagrammes ont été supprimés. Vous pouvez supprimer le numéro de révision en utilisant le bouton **Remise à zéro** pour réinitialiser l'historique de votre VI avant de supprimer le diagramme.

Impression des informations de l'historique

Lorsque vous imprimez un VI, vous pouvez inclure l'historique et le numéro de révision en sélectionnant **Documentation complète** dans la boîte de dialogue **Imprimer la documentation**. Si vous voulez changer le format d'impression pour y faire figurer uniquement les informations de l'historique, vous pouvez utiliser le bouton **Paramètres d'impression personnalisée** pour indiquer exactement ce que vous voulez imprimer. De plus, vous pouvez exporter les informations de l'historique vers un fichier en sélectionnant **Enregistrer les infos-texte** dans la boîte de dialogue **Imprimer la documentation**.

Paramétrage des options associées dans les boîtes de dialogue Configuration du VI et Préférences

Les boîtes de dialogue **Configuration du VI** et **Préférences** ont toutes deux des options que vous pouvez utiliser pour indiquer comment se comporte l'historique du VI.

Vous pouvez utiliser les options de **Configuration du VI** pour indiquer, pour un VI donné, si les informations de l'historique sont automatiquement enregistrées et à quel moment. Consultez la section *Options de documentation* du chapitre 6, *Configuration des VIs et sous-VIs*, pour plus d'informations sur ces paramètres.

Vous pouvez utiliser les options de la boîte de dialogue **Préférences** pour définir les paramètres par défaut pour l'historique des nouveaux VIs. Vous pouvez également utiliser cette boîte de dialogue pour indiquer un nom d'utilisateur à entrer dans les informations d'historique. Consultez la section *Préférences de l'historique* du chapitre 7, *Personnalisation de votre environnement*, pour plus d'informations sur ces paramètres.

Performances

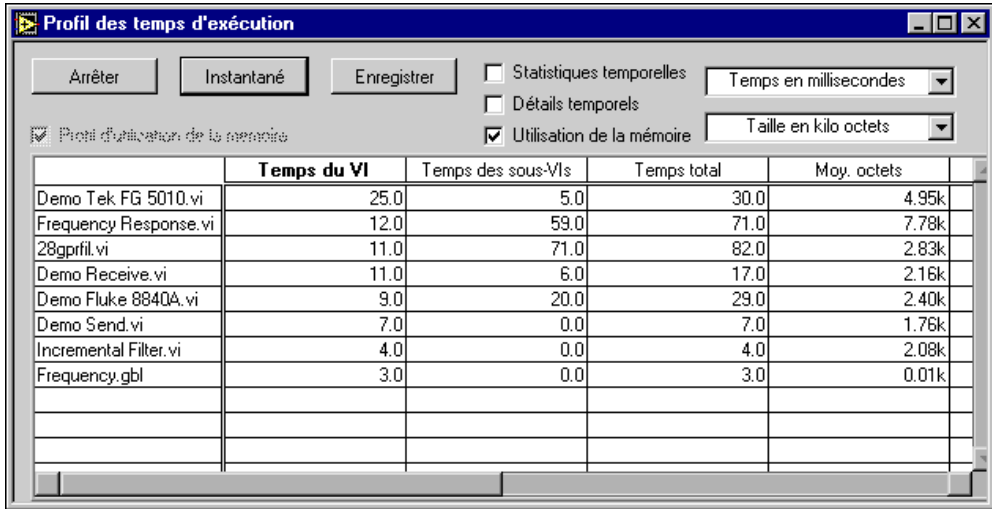
Ce chapitre est divisé en trois sections. La première section décrit l'optimiseur de performances, une fonction qui vous donne des informations sur le temps d'exécution de vos VIs et surveille les applications qui fonctionnent sur un ou plusieurs threads et sur plusieurs processeurs. La deuxième section décrit les facteurs qui affectent la vitesse d'exécution. La troisième section décrit les facteurs qui affectent l'usage de la mémoire.

Optimisation des performances des VIs

L'optimiseur de performances de VI est un outil puissant grâce auquel vous pouvez déterminer les parties de votre application qui consomment du temps et la façon dont est utilisée la mémoire. Ces informations sont précieuses pour trouver les foyers d'activité de votre application, vous permettant de faire un usage optimal des VIs les plus gourmands en temps. Un affichage tabulaire interactif vous indique l'usage de temps et de mémoire de chaque VI présent dans votre application. Chaque rangée du tableau contient des informations correspondant à un VI spécifique. Le temps utilisé par chaque VI est réparti entre plusieurs catégories et résumé sous forme de quelques chiffres statistiques. La fenêtre d'optimisation de performances calcule les temps minimum, maximum et moyen utilisés par exécution d'un VI. C'est ce qu'on appelle le profil des temps d'exécution.

Vous pouvez utiliser l'affichage tabulaire interactif pour visualiser l'ensemble ou une partie de ces informations, pour les classer en différentes catégories et pour examiner les performances des sous-VIs lorsqu'ils sont appelés à partir d'un VI spécifique.

Sélectionnez **Projet»Fenêtre d'optimisation** pour accéder à la fenêtre d'optimisation. L'illustration suivante montre un exemple d'une fenêtre déjà en service.



Il y a plusieurs choses à remarquer en ce qui concerne la fenêtre. D'abord, le recueil d'informations concernant l'usage de mémoire est optionnel, en raison du fait que ce processus peut ajouter une quantité non négligeable de temps machine au temps d'exécution de vos VIs. Vous devez choisir si vous voulez recueillir ces données avant de lancer l'exécution du Profil en cochant ou non, selon le cas, la case **Profil d'utilisation de la mémoire**. Cette case ne peut pas être modifiée une fois qu'une création de profil est en cours. Les boutons suivants sont disponibles dans la fenêtre d'optimisation.



- **Démarrer** : initie le recueil de données de performances. Il est préférable de commencer une création de profil avant d'entamer l'exécution de votre application. Ainsi, vous pouvez être sûr que vous mesurez seulement des exécutions complètes des VIs, et non des exécutions partielles.
- **Instantané** : visualise les données actuellement disponibles. Sous l'action de ce bouton, le système rassemble les données de tous les VIs en mémoire dans l'affichage tabulaire.
- **Enregistrer** : enregistre sur disque les données actuellement affichées, sous forme de fichier tableur à texte séparé par des tabulations. Ces données peuvent ensuite être visualisées dans un tableur ou par des VIs.

Visualisation des résultats

Vous pouvez choisir d'afficher seulement une partie des informations qui figurent dans le tableau. Certaines données de base restent toujours visibles, mais vous pouvez choisir d'afficher les statistiques, les détails et (si cette option est activée) l'utilisation de la mémoire en cochant les cases appropriées dans la fenêtre d'optimisation.

Des informations de performances sont également affichées pour les VIs globaux. Cependant, ces informations exigent parfois une interprétation légèrement différente, comme cela est décrit ci-dessous dans les sections spécifiques à chaque catégorie.

Les données indiquant les performances des sous-VIs lorsqu'ils sont appelés par un VI spécifique peuvent être visualisées en double-cliquant sur le nom de ce VI dans l'affichage tabulaire. En réponse à cette action, de nouvelles rangées apparaissent directement en dessous du nom du VI et contiennent des données de performances pour chacun de ses sous-VIs. Lorsque vous double-cliquez sur le nom d'un VI global, de nouvelles rangées apparaissent pour chacune des commandes individuelles de sa face-avant.

Vous pouvez classer les rangées de données dans l'affichage tabulaire en cliquant sur l'en-tête de colonne désiré. La colonne dictant le classement courant est indiquée par un en-tête en caractères gras.

Le temps d'exécution d'un VI ne correspond pas nécessairement au temps écoulé entre le début et la fin de l'exécution du VI. Ceci est dû au fait qu'un système d'exécution multithread peut entrelacer l'exécution de deux VIs ou plus. En outre, il y a une certaine quantité de temps machine qui n'est attribuée à aucun VI, comme le temps que met un utilisateur à répondre à une boîte de dialogue, le temps passé dans une fonction Attendre sur un diagramme ou le temps consacré à détecter les clics de la souris.

Les informations de base qui sont toujours visibles dans les trois premières colonnes du tableau de données de performances sont les suivantes :

- **Temps du VI** : temps total consacré à l'exécution effective du code de ce VI et à l'affichage de ses données, auquel est ajouté le temps passé par l'utilisateur à interagir avec les commandes de sa face-avant (le cas échéant). Cette quantité globale est répartie entre différentes sous-catégories dans les **Détails temporels**, décrits ci-dessous. Pour les VIs globaux, ce temps représente la durée totale passée à copier et à lire des données sur l'ensemble de leurs commandes. Pour obtenir des informations temporelles sur les commandes individuelles d'un VI global, vous pouvez double-cliquer sur le nom du VI global.
- **Temps des sous-VIs** : temps total utilisé par tous les sous-VIs de ce VI. C'est la somme du temps de VI (décrit ci-dessus) de tous les sous-VIs appelés par ce VI, ainsi que de celui des sous-VIs qu'ils ont appelés, etc.
- **Temps total** : somme des deux catégories ci-dessus, donnant le temps total écoulé.

Informations temporelles

Lorsque la case **Statistiques temporelles** est cochée, les colonnes suivantes deviennent visibles dans l'affichage tabulaire :

- **Exécutions** : nombre de fois que l'exécution de ce VI a eu lieu. Pour les VIs globaux, cette valeur représente le nombre total d'accès à l'ensemble de leurs commandes.
- **Moyenne** : durée moyenne d'une exécution de ce VI. C'est simplement le temps du VI divisé par le nombre d'exécution.
- **Le plus rapide** : temps minimum qu'il a fallu au VI pour une exécution.
- **Le plus lent** : temps maximum qu'il a fallu au VI pour une exécution.

Lorsque la case **Détails temporels** est cochée, vous pouvez visualiser une répartition entre plusieurs catégories temporelles du temps utilisé par le VI. Dans le cas des VIs dont l'interface utilisateur est complexe, ces catégories peuvent vous aider à voir quelles opérations prennent le plus de temps.

- **Diagramme** : temps consacré uniquement à l'exécution du code généré pour le diagramme de ce VI.
- **Affichage** : temps consacré à la mise à jour des commandes de la face-avant de ce VI avec de nouvelles valeurs provenant du diagramme.

- **Traçage** : temps consacré à dessiner la face-avant de ce VI. Le temps de traçage comprend :
 - Le temps qu’il faut pour simplement dessiner une face-avant lorsque sa fenêtre vient juste d’être ouverte ou lorsqu’elle revient au premier plan après avoir été masquée par une autre fenêtre.
 - Le temps qui est théoriquement du temps d’affichage, dû à l’arrivée de nouvelles valeurs provenant du diagramme, mais survenant parce que la commande est transparente et/ou recouverte. Ces commandes doivent invalider leur zone écran lorsqu’elles reçoivent de nouvelles données du diagramme, afin que tout dans cette zone puisse être redessiné dans le bon ordre. Les autres commandes peuvent être immédiatement dessinées sur la face-avant lorsqu’elles reçoivent de nouvelles données du diagramme. Il y a davantage de temps machine qui est utilisé pour invalider et redessiner, et la majeure partie de celui-ci (mais pas la totalité) est comptabilisée dans le temps de Traçage.
- **Repérage souris** : temps consacré à suivre la souris pendant que l’utilisateur interagit avec la face-avant de ce VI. Celui-ci peut être considérable pour certains types d’opérations, par exemple pour faire un zoom avant ou arrière sur un graphe, pour sélectionner des éléments à partir d’un menu local ou pour sélectionner et taper du texte dans une commande.
- **Variables locales** : temps consacré à copier des données vers des variables locales, ou à lire des données à partir de variables locales, sur le diagramme. L’expérience avec les utilisateurs montre que ce temps peut parfois être considérable, en particulier lorsqu’il s’agit de volumes importants de données complexes.

Informations sur la mémoire

Lorsque la case **Utilisation de la mémoire** est cochée (rappelez-vous que, celle-ci n’est disponible que si la case **Profil d’utilisation de la mémoire** est cochée avant le début de la création du profil), vous pouvez visualiser des informations sur la façon dont vos VIs utilisent la mémoire. Ces valeurs sont une mesure de l’espace mémoire utilisé par le VI et ne comprennent pas les structures de données de support nécessaires pour tous les VIs. L’espace mémoire pour le VI contient non seulement les données explicitement utilisées par les commandes de la face-avant, mais aussi des buffers temporaires créés implicitement par le compilateur.

Les tailles de mémoire sont mesurées à la fin de l'exécution d'un VI, et il est possible qu'elles ne reflètent pas exactement son usage total. Par exemple, si un VI crée de grands tableaux au cours de son exécution mais réduit leur taille avant la fin de l'exécution du VI, les tailles affichées ne reflètent pas les tailles intermédiaires plus grandes.

Deux jeux de données sont affichés dans cette section : les données relatives au nombre d'octets utilisés et les données relatives au nombre de blocs utilisés. Un bloc est un segment contigu de mémoire utilisé pour enregistrer des données en un seul morceau. Par exemple, un tableau d'entiers peut avoir une longueur de plusieurs octets, mais il occupe seulement un bloc. Le système d'exécution utilise des blocs de mémoire indépendants pour les tableaux, les chaînes de caractères, les chemins et les images (à partir du Picture Control Toolkit). Un grand nombre de blocs dans la pile de mémoire de votre application peut provoquer une dégradation globale des performances (pas seulement de l'exécution).

Les catégories répertoriées sous Utilisation de la mémoire sont les suivantes :

- **Moy. octets** : nombre moyen d'octets utilisés par l'espace mémoire de ce VI par exécution.
- **Octets min.** : nombre minimum d'octets utilisés par l'espace mémoire de ce VI pour une exécution individuelle.
- **Octets max.** : nombre maximum d'octets utilisés par l'espace mémoire de ce VI pour une exécution individuelle.
- **Moy. blocs** : nombre moyen de blocs utilisés par l'espace mémoire de ce VI par exécution.
- **Blocs min.** : nombre minimum de blocs utilisés par l'espace mémoire de ce VI pour une exécution individuelle.
- **Blocs max.** : nombre maximum de blocs utilisés par l'espace mémoire de ce VI pour une exécution individuelle.

Vitesse d'exécution des VIs

Bien que le compilateur produise un code dont l'exécution est généralement très rapide, vous avez intérêt, lorsque vous travaillez sur des applications qui consomment beaucoup de temps, à faire tout ce que vous pouvez pour obtenir les meilleures performances de vos VIs. Cette section aborde les facteurs qui affectent la vitesse d'exécution et suggère certaines techniques de programmation pour vous aider à obtenir les meilleures performances possibles.

Examinez les éléments suivants pour déterminer les causes d'une lenteur éventuelle au niveau des performances :

- Entrée/Sortie (fichiers, GPIB, acquisition de données, réseau)
- Affichage à l'écran (commandes de grande taille, commandes qui se chevauchent, trop d'éléments affichés)
- Gestion de la mémoire (utilisation inefficace des tableaux et chaînes de caractères, structures de données inefficaces)

D'autres facteurs peuvent jouer, comme le temps machine consommé lors de l'exécution et lors des appels de sous-VIs, mais ceux-ci sont généralement minimales et ne constituent pas la cause principale d'une lenteur de l'exécution.

Entrée/Sortie

Les appels d'Entrée/Sortie consomment généralement beaucoup de temps machine. Il y a souvent un ordre de grandeur de différence entre le temps qu'ils prennent et le temps qu'il faut pour effectuer une opération de calcul. Par exemple, une simple opération de lecture du port série peut coûter plusieurs millisecondes de temps machine. Cette quantité de temps machine s'applique à n'importe quelle application qui utilise des ports série. La raison de ce temps machine est qu'un appel d'E/S exige de transférer des informations au travers de plusieurs niveaux d'un système d'exploitation.

La meilleure méthode pour réduire les excès de temps machine est de minimiser le nombre d'appels d'E/S que vous faites. Vos performances seront meilleures si vous pouvez structurer votre application de manière à transférer une grande quantité de données à chaque appel, au lieu de faire beaucoup d'appels d'E/S portant sur de plus petites quantités de données.

Par exemple, si vous créez un VI d'acquisition de données (NI-DAQ), vous avez deux options pour lire les données. Vous pouvez utiliser une fonction de transfert d'une donnée, telle que le VI "AI-Echantillonner une voie" (AI Sample Channel.vi), ou vous pouvez utiliser une fonction de transfert de plusieurs données, telle que le VI "AI-Acquérir un signal" (AI Acquire Waveform.vi). Si vous devez acquérir 100 points, utilisez le VI "AI-Echantillonner une voie" (AI Sample Channel.vi) au sein d'une boucle, avec une fonction Attendre pour établir le séquençement d'exécution. Ou bien, vous pouvez utiliser le VI "AI-Acquérir un signal" (AI Acquire Waveform.vi) avec une entrée indiquant que vous voulez 100 points.

Vous pouvez produire des fréquences d'échantillonnage de données beaucoup plus élevées et plus précises en utilisant le VI "AI-Acquérir un

signal” (AI Acquire Waveform.vi), étant donné qu’il utilise des horloges matérielles pour gérer l’échantillonnage des données. De plus, le temps machine pour le VI “AI-Acquérir un signal” (AI Acquire Waveform.vi) est approximativement le même que celui d’un seul appel au VI “AI-Echantillonner une voie” (AI Sample Channel.vi), alors qu’il transfère beaucoup plus de données.

Affichage à l’écran

La mise à jour des commandes sur une face-avant peut souvent être l’une des opérations les plus coûteuses en temps dans une application. Ceci est particulièrement vrai si vous utilisez certains affichages compliqués, comme des graphes ou des graphes déroulants.

Ce temps machine est minimisé, dans une certaine mesure, par le fait que la plupart des commandes sont “intelligentes”. Elles ne se redessinent pas lorsqu’elles reçoivent de nouvelles données si les nouvelles données sont les mêmes que les anciennes. Les graphes et les graphes déroulants font exception à cette règle.

Si la vitesse de dessin devient un problème, les meilleures solutions consistent à réduire le nombre de commandes que vous utilisez et à garder l’affichage aussi simple que possible. Dans le cas des graphes et graphes déroulants, vous pouvez désactiver la mise à l’échelle automatique, les repères d’échelle et les grilles pour accélérer l’affichage.

Si certaines de vos commandes sont recouvertes par d’autres objets, leur vitesse d’affichage est considérablement réduite. La raison pour ceci est que, si une commande est partiellement masquée, il y a plus de travail pour redessiner cette zone de l’écran. Si vous n’avez pas activé, parmi vos préférences, l’option **Rafraîchissement progressif**, vous pouvez observer un scintillement plus prononcé lorsque des commandes sont recouvertes.

Comme pour les autres types d’E/S, il y a une certaine quantité de temps machine fixe qui entre en jeu dans l’affichage d’une commande. Vous pouvez transmettre plusieurs points à la fois à un indicateur à l’aide de certaines commandes, telles que des graphes déroulants. Vous pouvez minimiser le nombre de mises à jour d’un graphe déroulant en transférant à chaque fois davantage de données vers celui-ci. Vous pouvez obtenir des vitesses d’affichage de données beaucoup plus élevées avec vos graphes déroulants si vous recueillez vos données dans des tableaux pour afficher plusieurs points à la fois, au lieu d’afficher chaque point à mesure qu’il arrive.

Lorsque vous créez des sous-VIs dont la face-avant est fermée pendant l'exécution, vous n'avez pas à vous soucier du temps machine consacré à l'affichage. Si la face-avant est fermée, le temps machine pour le dessin des commandes est nul, donc les graphes ne coûtent pas plus de temps que les tableaux.

Les commandes et indicateurs ont un menu local d'affichage synchrone qui, dans les systèmes multithread, contrôle si les mises à jour peuvent ou non être différées. Sur un système d'exécution à un thread, cet élément n'a aucun effet. Cependant, si vous activez ou désactivez cet élément dans des VIs dans la version à un thread, ces changements affectent le comportement des mises à jour si vous chargez ces VIs dans un système multithread.

Par défaut, cet élément est désactivé, ce qui signifie que lorsque le système d'exécution transmet des données aux commandes et indicateurs de la face-avant, il peut transmettre les données à la commande ou à l'indicateur et poursuivre immédiatement son exécution. Après cela, le système d'interface utilisateur remarque que la commande ou l'indicateur a besoin d'être mis à jour, et il le redessine pour afficher les nouvelles données. Si le système d'exécution essaie de mettre à jour la commande plusieurs fois au cours d'un bref intervalle de temps, il est possible que vous ne voyiez pas certaines des mises à jour intermédiaires.

Dans de nombreuses applications, cela peut considérablement accélérer l'exécution sans affecter ce que l'utilisateur peut voir. Par exemple, même lorsque cet élément est désactivé, vous pouvez mettre à jour un booléen des centaines de fois par seconde, avec plus de mises à jour que l'œil humain peut effectivement discerner. Un affichage asynchrone permet au système d'exécution de consacrer davantage de temps à l'exécution des VIs, le rythme des mises à jour étant automatiquement ralenti par le thread de l'interface utilisateur.

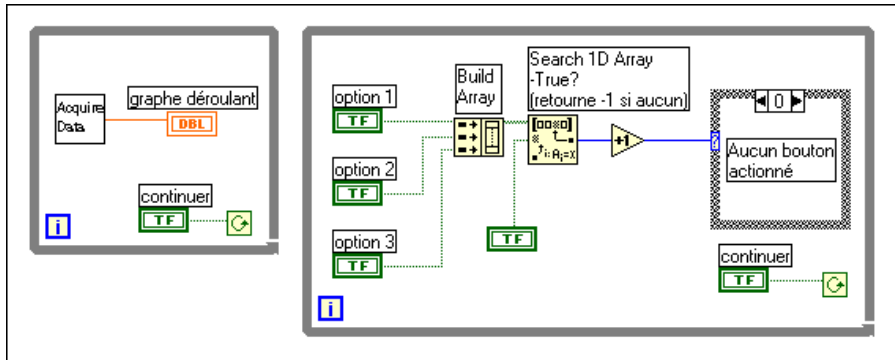
Si vous voulez éviter ce type d'affichage, vous pouvez activer l'affichage synchrone.

Autres questions

Diagrammes parallèles

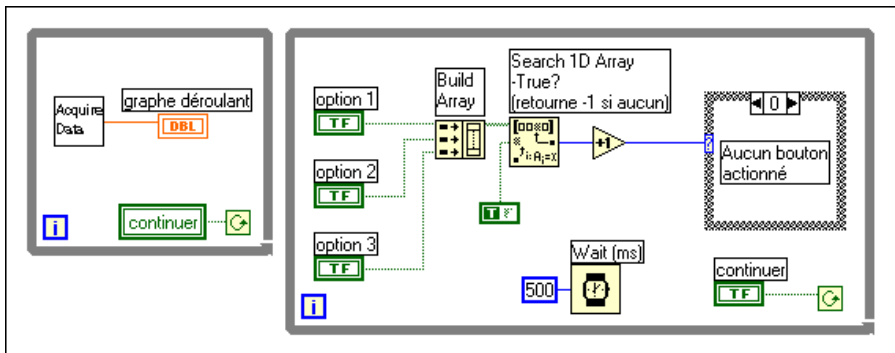
Lorsque vous avez plusieurs diagrammes qui fonctionnent en parallèle, le système d'exécution alterne périodiquement entre eux. Si certaines de ces boucles sont moins importantes que d'autres, utilisez la fonction Attendre pour garantir que les boucles moins importantes consomment moins de temps.

Prenez, par exemple, le cas du diagramme suivant :



Il y a deux boucles en parallèle. Une des boucles effectue l'acquisition des données et doit être exécutée aussi fréquemment que possible. L'autre boucle contrôle les entrées de l'utilisateur. En raison de la façon dont ce programme est structuré, les deux boucles reçoivent autant de temps. La boucle surveillant les actions de l'utilisateur peut être exécutée plusieurs fois par seconde.

En pratique, il est généralement acceptable que la boucle contrôlant le bouton s'exécute seulement une fois toutes les demi-secondes, ou même moins souvent. En appelant la fonction Attendre (ms) dans la boucle d'interface utilisateur, vous allouez beaucoup plus de temps à l'autre boucle.



Temps machine d'accès aux sous-VIs

Lorsque vous appelez un sous-VI, il y a une certaine quantité de temps machine qui résulte de cet appel. Ce temps machine est relativement minime (de l'ordre de quelques dizaines de microsecondes), surtout si on le compare au temps machine que coûtent les opérations d'E/S et d'affichage, qui peut aller de quelques millisecondes à plusieurs dizaines de millisecondes.

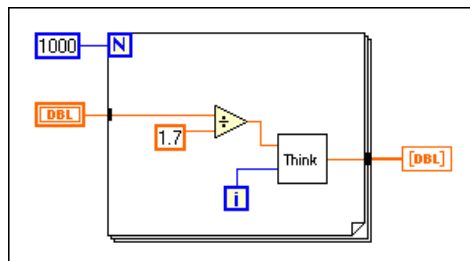
Cependant, dans certains cas, ce temps machine peut s'accumuler. Par exemple, si vous appelez un sous-VI 10 000 fois dans une boucle, le temps machine peut atteindre une valeur non négligeable. Dans ce cas, il peut être intéressant d'envisager d'inclure la boucle dans le sous-VI.

Une autre option que vous pouvez envisager consiste à transformer certains sous-VIs en sous-programmes (à l'aide de l'option **Priorité de configuration du VI**). Lorsqu'un VI est désigné comme sous-programme, le système d'exécution minimise le temps machine que consomme un appel au sous-VI. Cependant, il y a quelques contreparties. Les sous-programmes ne peuvent pas afficher les données des faces-avant (le G ne peut ni lire ni écrire des données dans les commandes de la face-avant des sous-programmes), ils ne peuvent contenir ni fonction de cadencement d'exécution ni fonction de boîte de dialogue, et ils ne peuvent pas fonctionner en multitâche avec d'autres VIs. Les sous-programmes sont des tâches courtes et fréquemment exécutées, et leur utilisation la plus appropriée est généralement avec des VIs qui ne nécessitent pas d'interaction avec l'utilisateur. Pour plus d'informations, consultez le chapitre 26, *Comprendre le système d'exécution du G*.

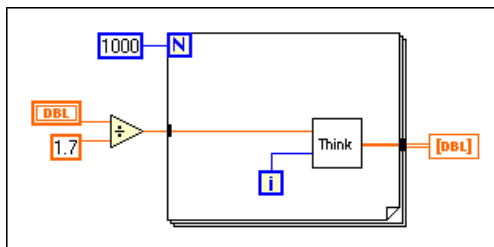
Calculs superflus dans des boucles

Évitez de mettre des calculs dans des boucles si le calcul produit la même valeur à chaque itération. De préférence, sortez le calcul de la boucle et transmettez le résultat dans la boucle.

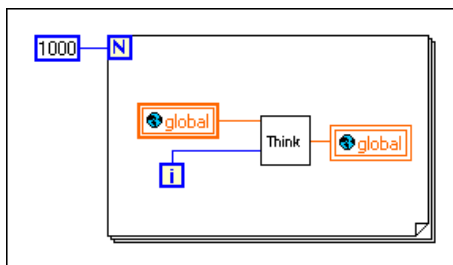
Par exemple, examinez le diagramme suivant :



Le résultat de la division est le même à chaque itération de la boucle ; par conséquent, vous pouvez augmenter les performances en sortant la division de la boucle, comme cela est représenté dans l'illustration suivante :

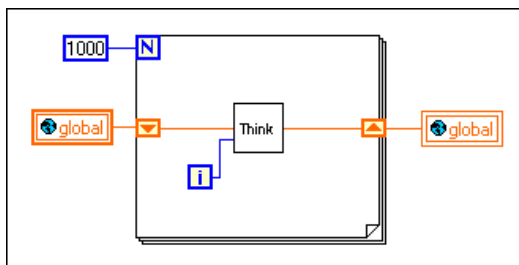


Maintenant, reportez-vous au diagramme de l'illustration suivante :

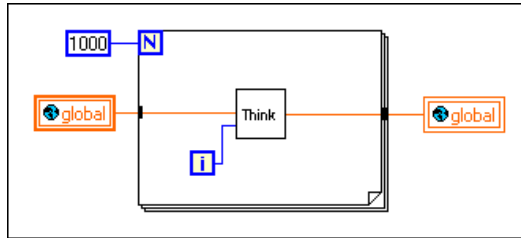


Si vous savez que la valeur de la variable globale ne va pas être modifiée pendant l'exécution de cette boucle par un autre diagramme ou VI s'exécutant en parallèle, ce diagramme gaspille du temps à lire et à écrire la variable globale à chaque itération de la boucle.

Si vous n'avez pas besoin que la variable globale soit lue ou écrite par un autre diagramme pendant l'exécution de cette boucle, vous pouvez utiliser le diagramme suivant :



Notez que les registres à décalage doivent transmettre la nouvelle valeur du sous-VI à l'itération suivante de la boucle. Le diagramme suivant est une erreur courante que commettent souvent les utilisateurs novices. Etant donné qu'il n'y a pas de registre à décalage, les résultats du sous-VI ne sont jamais renvoyés au sous-VI en tant que nouvelle valeur d'entrée.



Utilisation de la mémoire par les VIs

Le G règle un grand nombre de détails dont vous devez normalement vous soucier avec un langage de programmation conventionnel. Une des principales difficultés d'un langage conventionnel est l'utilisation de la mémoire. Avec un langage conventionnel, c'est vous, le programmeur, qui devez prendre soin d'allouer de la mémoire avant de l'utiliser et de la libérer une fois que vous avez fini. Vous devez aussi faire particulièrement attention à ne pas écrire accidentellement au-delà de la mémoire que vous avez allouée au départ. Une des plus grosses erreurs que font les programmeurs avec les langages textuels conventionnels est de ne pas allouer de mémoire ou de ne pas en allouer assez. En outre, une allocation de mémoire insuffisante est un problème difficile à déceler.

Le principe de flux de données sur lequel repose le G élimine beaucoup des difficultés inhérentes à la gestion de la mémoire. Avec le G, vous n'allouez pas de variables, et vous n'assignez pas non plus de valeurs à celles-ci ou à partir de celles-ci. A la place, vous créez un diagramme avec des connexions représentant le cheminement des données.

Les fonctions qui génèrent des données s'occupent d'allouer de la mémoire pour ces données. Lorsque les données ne sont plus utilisées, la mémoire associée est libérée. Lorsque vous ajoutez de nouvelles informations à un tableau ou à une chaîne de caractères, une quantité suffisante de mémoire est allouée automatiquement pour recevoir les nouvelles informations.

Cette gestion automatique de la mémoire est l'un des principaux avantages du G. Cependant, dans la mesure où c'est automatique, vous avez moins de contrôle sur le moment auquel cela se produit. Si votre programme travaille

avec de gros volumes de données, il est important d'avoir une idée du moment auquel a lieu l'allocation de mémoire. Une bonne compréhension des principes utilisés peut se traduire par des programmes qui nécessitent beaucoup moins de mémoire. De plus, comprendre comment minimiser l'utilisation de la mémoire peut aussi vous aider à augmenter la vitesse d'exécution des VIs, parce que l'allocation de mémoire et la copie de données peuvent prendre un temps considérable.

Mémoire virtuelle

Si vous avez une machine qui dispose d'une quantité de mémoire limitée, vous pouvez envisager d'utiliser de la mémoire virtuelle pour augmenter la quantité de mémoire disponible pour les applications. La mémoire virtuelle est une capacité de votre système d'exploitation, lui permettant d'utiliser de l'espace disque disponible comme mémoire RAM. Si vous allouez une grande quantité de mémoire virtuelle, les applications perçoivent cela comme de la mémoire généralement disponible pour un stockage de données.

Pour les applications, il importe peu que votre mémoire soit de la mémoire RAM réelle ou de la mémoire virtuelle, le système d'exploitation dissimulant le fait que la mémoire est virtuelle. La principale différence est la vitesse. Avec de la mémoire virtuelle, il est possible que vous ressentiez occasionnellement des baisses de performances lorsque le système d'exploitation échange de la mémoire avec le disque. La mémoire virtuelle peut vous aider à exécuter des applications plus volumineuses, mais elle n'est probablement pas appropriée pour des applications qui ont des contraintes temporelles strictes.

Mémoire du Macintosh

Lorsque vous lancez une application sur Macintosh, le système lui alloue un bloc de mémoire unique, à partir duquel toute la mémoire est allouée. Lorsque vous chargez des VIs, les composantes de ces derniers sont chargées dans cette mémoire. De même, lorsque vous exécutez un VI, toute la mémoire qu'il manipule est allouée à partir de ce bloc unique de mémoire.

Vous pouvez configurer la quantité de mémoire que le système alloue au moment du lancement à l'aide de la commande **Fichier»Obtenir des infos** du Finder. N'oubliez pas que si votre application se retrouve à court de mémoire, elle ne peut pas augmenter la taille de ce capital de mémoire. Par conséquent, donnez à ce paramètre une valeur aussi élevée que possible. Si vous avez une machine de 16 Mo, tenez compte des applications que vous

voulez exécuter en parallèle. Si vous n'envisagez pas d'exécuter d'autres applications, réglez la mémoire sur la valeur la plus élevée possible.

Gestion de la mémoire par les composantes des VIs

Un VI est constitué de quatre composantes majeures :

- La face-avant
- Le diagramme
- Le code (diagramme compilé en code machine)
- Les données (valeurs des commandes et indicateurs, données par défaut, constantes des diagrammes, et ainsi de suite).

Lors du chargement d'un VI, la face-avant, le code (s'il correspond à la plate-forme) et les données pour le VI sont chargés en mémoire. Si le VI a besoin d'être recompilé à cause d'un changement de plate-forme ou d'un changement au niveau de l'interface d'un sous-VI, le diagramme est lui aussi chargé en mémoire.

Le VI charge également le code et l'espace mémoire de ses sous-VIs en mémoire. Dans certaines circonstances, la face-avant de certains sous-VIs peut elle aussi être chargée en mémoire. Ceci peut se produire, par exemple, si le sous-VI utilise des attribute nodes, étant donné que les attribute nodes manipulent des informations d'état pour les commandes de la face-avant. De plus amples informations sur les sous-VIs et leur face-avant sont fournies plus loin dans ce chapitre.

Un point important de l'organisation des composantes des VIs est que vous n'utilisez généralement pas beaucoup plus de mémoire lorsque vous convertissez une section de votre VI en sous-VI. Si vous créez un VI unique, énorme et ne comportant aucun sous-VI, vous vous retrouvez avec la face-avant, le code et les données de ce VI principal en mémoire. Cependant, si vous décomposez le VI en sous-VIs, le code pour le VI principal est plus petit, et le code et les données des sous-VIs résident en mémoire. Dans certains cas, vous pourrez constater une réduction de l'utilisation de la mémoire en exécution. Ce concept est expliqué plus loin dans ce chapitre, dans la section *Contrôle de l'utilisation de la mémoire*.

Vous pourrez observer également que l'édition des VIs volumineux prend plus de temps. Vous ne rencontrez pas autant ce problème si vous décomposez votre VI en sous-VIs, parce que l'éditeur peut gérer plus efficacement les VIs de plus petite taille. De plus, une organisation plus hiérarchique des VIs est généralement plus facile à entretenir et à lire.

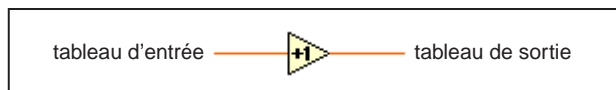
 **Remarque**

Si la face-avant ou le diagramme d'un VI donné est beaucoup plus grand que votre écran, vous pouvez le décomposer en sous-VIs pour le rendre plus accessible.

Programmation par flux de données et buffers de données

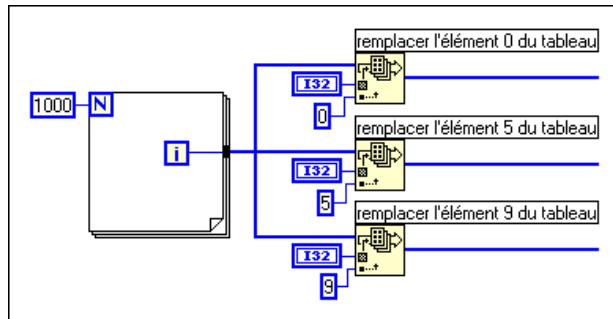
En programmation par flux de données, vous n'utilisez généralement pas de variables. Les modèles de flux de données décrivent normalement des nœuds qui acceptent des entrées et produisent des sorties. Une mise en œuvre littérale de ce modèle donne lieu à des applications qui peuvent utiliser de très grandes quantités de mémoire et avoir des performances décevantes. Chaque fonction produit une copie des données pour chaque destination vers laquelle sont envoyées les sorties. Le compilateur G améliore cette approche en essayant de déterminer à quel moment la mémoire peut être réutilisée et en examinant les destinations d'une sortie pour déterminer s'il est nécessaire de faire des copies pour chaque terminal.

Par exemple, avec une approche plus traditionnelle du compilateur, le diagramme suivant utilise deux blocs de mémoire de données, un pour l'entrée et un autre pour la sortie.



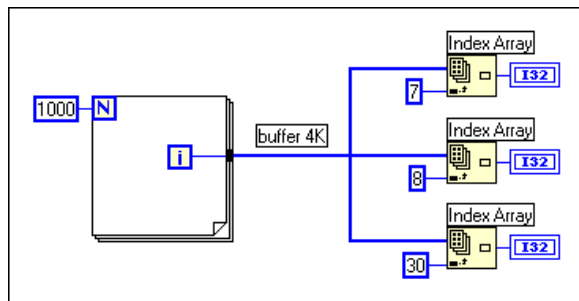
Le tableau d'entrée et le tableau de sortie contiennent le même nombre d'éléments, et le type de données pour les deux tableaux est le même. Considérez maintenant le tableau d'entrée comme un buffer de données. Au lieu de créer un nouveau buffer pour la sortie, le compilateur réutilise le buffer d'entrée. Cela économise de la mémoire et, de plus, cela permet une exécution plus rapide dans la mesure où il n'y a pas besoin de procéder à une allocation de mémoire pendant l'exécution.

Cependant, le compilateur ne peut pas réutiliser les buffers de mémoire dans tous les cas, comme le montre l'illustration suivante :



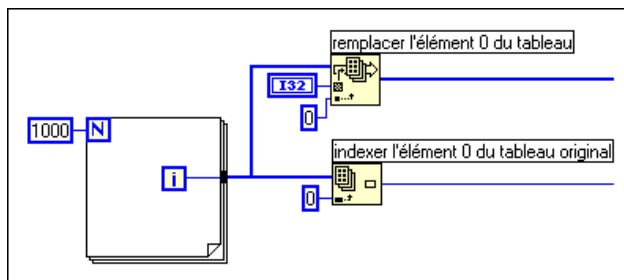
Des données provenant d'une source unique sont transmises vers plusieurs destinations. Les destinations veulent modifier les données pour produire des tableaux de résultats. Dans ce cas, le compilateur crée de nouveaux buffers de données pour deux des fonctions et copie les données du tableau dans les buffers. Par conséquent, une des fonctions réutilise le tableau d'entrée, et les autres pas. Ce diagramme utilise environ 12 Ko (4 Ko pour le tableau original et 4 Ko pour chacun des deux buffers de données supplémentaires).

Maintenant, examinez le diagramme suivant :



Comme précédemment, les branches d'entrée vers les trois fonctions sont les mêmes. Cependant, dans ce cas, aucune d'entre elles n'a besoin de modifier les données. Si vous transmettez des données vers plusieurs endroits, et si chacun d'eux lit les données sans les modifier, le G ne fait pas de copie des données. Ce diagramme utilise environ 4 Ko de données.

Enfin, considérez le diagramme suivant :



Dans ce cas, les entrées sont envoyées vers deux fonctions, et l'une d'elles veut modifier les données. Il n'y a aucune dépendance entre les deux fonctions. Par conséquent, vous pouvez penser qu'au moins une copie doit être réalisée pour que la fonction qui remplace les éléments du tableau puisse modifier sans risque les données. Cependant, dans ce cas, le compilateur organise l'exécution des fonctions de manière à exécuter en premier celle qui veut lire les données, et en dernier celle qui veut modifier les données. De cette façon, la fonction "Remplacer un élément d'un tableau" réutilise le buffer du tableau d'entrée sans générer de copie du tableau. Si l'ordonnancement des nœuds est important, rendez-le explicite en utilisant soit une séquence, soit une sortie d'un nœud pour l'entrée d'un autre.

En pratique, l'analyse des diagrammes par le compilateur n'est pas parfaite. Dans certains cas, le compilateur peut ne pas être capable de déterminer la méthode optimale pour réutiliser la mémoire du diagramme.

Contrôle de l'utilisation de la mémoire

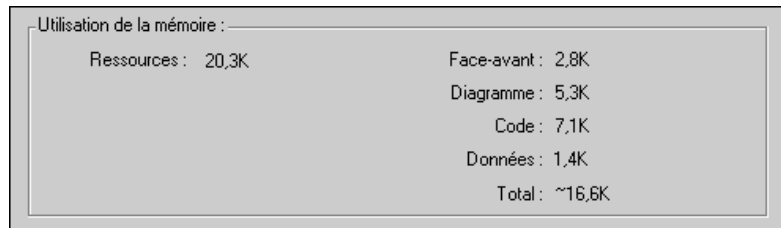
Il y a plusieurs méthodes pour déterminer l'utilisation de la mémoire.

Si vous sélectionnez **Aide»A propos de LabVIEW...** (si vous utilisez LabVIEW) ou **Help»About BridgeVIEW...** (si vous utilisez BridgeVIEW), vous pouvez voir des statistiques qui résument la quantité totale de mémoire utilisée par votre application. Cette mémoire comprend la mémoire pour les VIs ainsi que la mémoire utilisée par l'application. Vous pouvez vérifier cette valeur avant et après l'exécution d'un ensemble de VIs pour obtenir une idée approximative de la quantité de mémoire qu'utilisent les VIs.

Vous pouvez obtenir une image de l'utilisation dynamique de mémoire par vos VIs avec l'optimiseur de performances. Il enregistre des statistiques sur le nombre minimum, le nombre maximum et le nombre moyen d'octets et

de blocs utilisés par chaque VI par exécution. Consultez la section *Optimisation des performances des VIs*, au début de ce chapitre, pour plus de détails.

Comme le montre l'illustration suivante, vous pouvez utiliser **Fenêtres» Infos sur le VI...** pour obtenir la répartition détaillée de l'utilisation de mémoire pour un VI donné. La colonne de gauche de ces informations résume l'utilisation du disque et la colonne de droite indique la quantité de mémoire RAM qui est actuellement utilisée pour diverses composantes du VI. Notez que ces statistiques ne comprennent pas l'utilisation de mémoire des sous-VIs.



Une quatrième méthode pour déterminer l'utilisation de la mémoire est d'employer un VI appelé "Contrôleur de mémoire.vi" (Memory Monitor.vi), qui se trouve dans `memmon.11b` à l'intérieur de la bibliothèque `examples`. Ce VI utilise les fonctions du VI Serveur pour déterminer l'utilisation de mémoire pour tous les VIs en mémoire.

Règles pour une meilleure utilisation de la mémoire

Ce qu'il faut retenir de la section précédente c'est que le compilateur essaie de réutiliser la mémoire de façon intelligente. Les règles utilisées pour déterminer à quel moment le compilateur peut réutiliser de la mémoire et à quel moment il ne le peut pas sont relativement complexes ; elles sont évoquées à la fin de ce chapitre. En pratique, les règles suivantes peuvent vous aider à créer des VIs qui utilisent la mémoire de façon efficace :

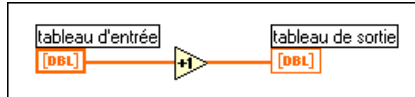
- Décomposer un VI en sous-VIs n'a généralement pas un impact négatif sur l'utilisation de la mémoire. En fait, dans de nombreux cas, l'utilisation de la mémoire est améliorée, étant donné que le système d'exécution peut récupérer la mémoire des données du sous-VI lorsque le sous-VI n'est pas en cours d'exécution.
- Ne vous souciez pas trop des copies de valeurs scalaires ; il faut beaucoup de scalaires pour ressentir un effet négatif sur l'utilisation de la mémoire.

- N'utilisez pas trop de variables globales et locales lorsque vous travaillez avec des tableaux ou des chaînes de caractères ; la lecture d'une variable globale ou locale génère une copie de la valeur ou de la variable.
- Sur les faces-avant ouvertes, n'affichez pas de grands tableaux ou chaînes de caractères, à moins que cela ne soit nécessaire. Les indicateurs des faces-avant ouvertes conservent une copie des données qu'ils affichent.
- Si la face-avant d'un sous-VI n'est pas appelée à être affichée, ne laissez pas les attribut nodes inutilisés sur le sous-VI. Les attribut nodes obligent la face-avant d'un sous-VI à rester en mémoire, ce qui peut donner lieu à une utilisation de mémoire superflue.
- N'utilisez pas de type de données de taille non définie dans les VIs pour lesquels le temps d'exécution et l'espace mémoire sont critiques. La face-avant du sous-VI doit être chargée pour vérifier la gamme de données, et des copies supplémentaires des données sont faites pour les commandes et les indicateurs du sous-VI.
- Lors de la création de diagrammes, faites attention aux endroits où la taille d'une entrée est différente de la taille d'une sortie. Par exemple, si vous voyez des endroits où vous augmentez fréquemment la taille d'un tableau ou d'une chaîne de caractères à l'aide des fonctions "Construire un tableau" ou "Concaténer des chaînes de caractères", vous générez des copies des données.
- Utilisez des types de données compatibles pour les tableaux et faites attention aux points de coercion lorsque vous transmettez des données à des sous-VIs et à des fonctions ; lorsque vous changez de type de données, le système d'exécution fait une copie des données.
- N'utilisez pas de types de données hiérarchiques compliqués (par exemple, tableaux de clusters contenant de grands tableaux ou chaînes de caractères ou clusters contenant de grands tableaux ou chaînes de caractères). Vous utilisez ainsi plus de mémoire et les performances en souffrent. Consultez la section *Développement de structures de données efficaces*, plus loin dans ce chapitre, pour plus de détails et suggestions de tactiques pour choisir vos types de données.

Gestion de mémoire dans les faces-avant

Lorsqu'une face-avant est ouverte, les commandes et indicateurs gardent leurs propres copies privées des données qu'ils affichent.

L'illustration suivante montre la fonction d'incrément, à laquelle s'ajoutent une commande et un indicateur de face-avant.



Lorsque vous exécutez le VI, les données de la commande de la face-avant sont transmises au diagramme. La fonction d’incrémation réutilise le buffer d’entrée. L’indicateur fait alors une copie des données pour les besoins de l’affichage. Par conséquent, il y a trois copies du buffer.

Cette protection des données de la commande de la face-avant évite que vous entriez des données dans une commande, vous exécutiez le VI associé et que vous voyiez les données changer dans la commande à mesure qu’elles sont transmises aux nœuds suivants. De même, les données sont protégées dans le cas des indicateurs, afin qu’ils puissent afficher de façon fiable leur contenu jusqu’à ce qu’ils reçoivent de nouvelles données.

Avec les sous-VIs, vous pouvez utiliser des commandes et des indicateurs en tant qu’entrées et sorties. Le système d’exécution fait une copie des données des commandes et indicateurs du sous-VI dans les conditions suivantes :

- La face-avant est en mémoire — ceci peut se produire pour n’importe laquelle des raisons suivantes :
 - La face-avant est ouverte.
 - Le VI a été modifié, mais pas enregistré (toutes les composantes du VI restent en mémoire jusqu’à l’enregistrement du VI).
 - La face-avant utilise la fonction d’impression des données.
 - Le diagramme utilise des attribute nodes.
- Le VI utilise des variables locales.
- La face-avant utilise la fonction d’enregistrement des données.
- Une commande utilise un type de données de taille non définie.

Quelques-unes de ces raisons ne sont pas évidentes et nécessitent de plus amples explications.

Une raison porte sur les attributs tels que l’historique de graphe déroulant. Pour qu’un attribute node soit capable de lire l’historique d’un graphe déroulant dans des sous-VIs dont la face-avant est fermée, la commande ou l’indicateur doit afficher les données qui lui sont transmises. Etant donné qu’il y a de nombreux autres attributs comme celui-ci, le système d’exécution garde la face-avant d’un sous-VI en mémoire si ce sous-VI utilise des attribute nodes.

Si une face-avant utilise la fonction d'enregistrement ou d'impression des données de la face-avant, les commandes et les indicateurs gardent des copies de leurs données. De plus, les faces-avant sont gardées en mémoire pour l'impression des données, afin que la face-avant puisse être imprimée.

Si un VI utilise un type de données de taille non définie, des données sont copiées et lues dans chaque commande et indicateur de la face-avant. Les valeurs de la face-avant sont conservées, afin que la face-avant puisse être affichée si des données sortent de la gamme autorisée. N'utilisez pas de type de données de taille non définie si la mémoire et la vitesse sont des préoccupations importantes.

Souvenez-vous que, si vous configurez un sous-VI pour qu'il affiche sa face-avant lorsqu'il est appelé, à l'aide des options de configuration de VI ou de sous-VI, la face-avant est chargée en mémoire lorsque le sous-VI est appelé. Si vous sélectionnez l'élément **Fermer après l'exécution si fermé à l'origine**, la face-avant est retirée de la mémoire lorsque l'exécution du sous-VI est terminée.

Les sous-VIs peuvent réutiliser la mémoire des données

En général, un sous-VI peut utiliser des buffers de données provenant de son VI appelant aussi facilement que si les diagrammes du sous-VI étaient dupliqués au niveau principal. Dans la plupart des cas, vous n'utilisez pas plus de mémoire si vous convertissez une section de votre diagramme en sous-VI. Pour les VIs qui ont des besoins spéciaux en termes d'affichage, comme cela est décrit dans la section précédente, il peut y avoir une certaine utilisation de mémoire supplémentaire pour les faces-avant et les commandes.

Les variables locales ne peuvent pas réutiliser la mémoire des données

Lorsque vous créez des sous-VIs, vous créez un cadre connecteur qui décrit comment les données sont transférées vers le sous-VI et à partir de celui-ci. Les buffers de données qui proviennent des terminaux connectés à un cadre connecteur peuvent réutiliser des buffers de données provenant des VIs appelants. Les variables locales ne peuvent pas faire cela. Aussi, lorsque vous lisez une variable locale, vous créez un nouveau buffer pour les données de la commande qui lui est associée.

Si vous utilisez des variables locales pour transférer de grandes quantités de données d'un endroit à un autre sur le diagramme, vous utilisez

généralement plus de mémoire, et vous avez par conséquent une vitesse d'exécution plus lente, que si vous pouvez transférer les données à l'aide d'un fil de liaison.

Les variables globales gardent toujours des copies de leurs données

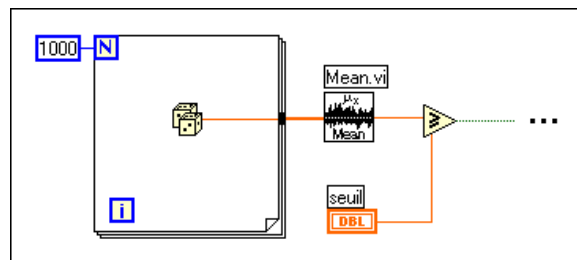
Lorsque vous lisez une variable globale, vous créez une copie des données stockées dans cette variable globale.

Lorsque vous manipulez de grands tableaux et chaînes de caractères, le temps et la mémoire requis pour manipuler des variables globales peuvent être considérables. Ceci est particulièrement inefficace avec des tableaux si vous voulez seulement modifier un seul élément du tableau, puis enregistrer le tableau tout entier. Si vous lisez la variable globale à plusieurs endroits dans votre diagramme, cela peut donner lieu à la création de plusieurs buffers de mémoire.

Une technique pour minimiser l'utilisation de mémoire dans ce cas consiste à utiliser un registre à décalage non initialisé dans un sous-VI pour stocker les données. Cette technique peut combiner l'aspect compact d'une variable globale à l'efficacité d'un registre à décalage. Voir *Etude de cas 2 : table globale de types de données mélangés*, plus loin dans ce chapitre, pour plus d'informations.

Comprendre à quel moment la mémoire est libérée

Considérez le diagramme suivant :



Une fois l'exécution du VI Moyenne (Mean.vi) terminée, le tableau de données n'est plus nécessaire. Etant donné qu'il peut être très compliqué de déterminer à quel moment les données ne sont plus nécessaires dans de grands diagrammes, l'exécution ne libère pas les buffers de données d'un VI particulier au cours de son exécution.

Sur Macintosh, si le système d'exécution n'a plus beaucoup de mémoire, il libère des buffers de données utilisés par un VI qui n'est pas actuellement en cours d'exécution. Le système d'exécution ne libère pas la mémoire utilisée par les commandes et indicateurs de la face-avant, les variables globales ou les registres à décalage non initialisés.

Maintenant, supposons que le VI décrit précédemment soit un sous-VI d'un VI plus grand. Le tableau de données est créé et utilisé uniquement dans le sous-VI. Sur Macintosh, si le sous-VI n'est pas en cours d'exécution et si le système n'a plus beaucoup de mémoire, il peut libérer les données dans le sous-VI. Ceci représente un cas dans lequel l'emploi de sous-VIs peut réduire l'utilisation de mémoire.

Sur les plates-formes Windows et Unix, les buffers de données ne sont normalement pas libérés, à moins qu'un VI ne soit fermé et retiré de la mémoire. La mémoire est allouée par le système d'exploitation selon les besoins et la mémoire virtuelle fonctionne bien sur ces plates-formes. A cause de la fragmentation, l'application peut sembler utiliser plus de mémoire qu'elle n'en utilise réellement. Lorsque de la mémoire est allouée et libérée, l'application essaie de consolider l'utilisation de mémoire, afin de pouvoir rendre les blocs non utilisés au système d'exploitation.

Sur toutes les plates-formes, vous pouvez activer en option une préférence appelée "Libérer la mémoire dès que possible". Lorsque cet élément est coché, les sous-VIs libèrent immédiatement la mémoire une fois que leur exécution est terminée. Cela peut être utile du point de vue de l'utilisation de la mémoire, mais cela ralentit considérablement les performances.

Déterminer à quel moment les sorties peuvent réutiliser les buffers d'entrée

Si une sortie a la même taille et le même type de données qu'une entrée, et si l'entrée n'est pas requise ailleurs, la sortie peut réutiliser le buffer d'entrée. Comme cela a été mentionné préalablement, dans certains cas, même lorsqu'une entrée est utilisée ailleurs, le compilateur et le système d'exécution peuvent arranger l'ordre d'exécution de manière à ce qu'il soit possible de réutiliser l'entrée en tant que buffer de sortie. Cependant, les règles appliquées pour cela sont complexes. Ne comptez pas dessus.

Types de données compatibles

Si une entrée et une sortie ont des types de données différents, la sortie ne peut pas réutiliser l'entrée. Par exemple, si vous ajoutez un entier 32 bits à un entier 16 bits, vous voyez un point de coercition qui indique que l'entier

16 bits est converti en entier 32 bits. L'entrée, en tant qu'entier 32 bits, peut être utilisable pour le buffer de sortie, en supposant qu'elle remplit toutes les autres conditions (par exemple, que l'entier 32 bits ne soit pas réutilisé ailleurs).

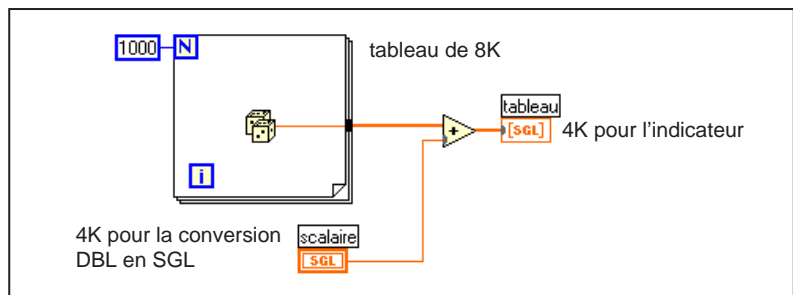
De plus, pour les sous-VIs et beaucoup de fonctions, les points de coercion impliquent une conversion du type de données. En général, le compilateur crée un nouveau buffer pour les données converties.

Pour minimiser l'utilisation de mémoire, utilisez autant que possible des types de données compatibles. Ceci réduit le nombre de copies de données créées pour promouvoir les données en taille. Le fait d'utiliser des types de données compatibles rend également le compilateur plus flexible pour déterminer à quel moment les buffers de données peuvent être réutilisés.

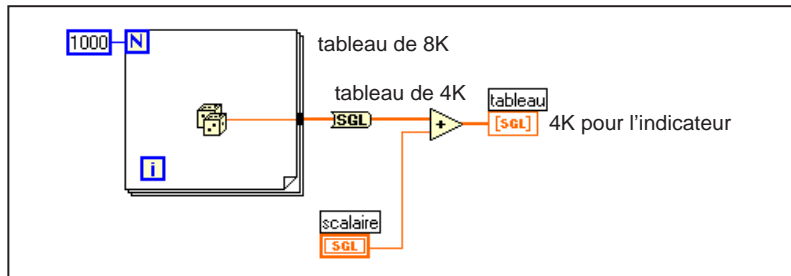
Dans certaines applications, vous pouvez envisager d'utiliser des types de données plus petits. Par exemple, vous pouvez envisager d'utiliser des nombres en simple précision, de quatre octets, au lieu de nombres en double précision, de huit octets. Cependant, prenez soigneusement en compte les types de données qui sont attendus par les sous-VIs que vous êtes susceptible d'appeler, parce que vous avez intérêt à éviter les conversions inutiles.

Comment générer des données du type voulu

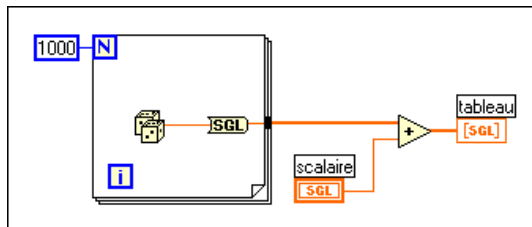
Reportez-vous à l'exemple suivant, dans lequel un tableau de 1000 valeurs aléatoires est créé et ajouté à un scalaire. Le point de coercion au niveau de la fonction Ajouter apparaît parce que les données aléatoires sont en double précision, alors que le scalaire est en simple précision. Le scalaire est promu en double précision avant l'addition. Les données résultantes sont alors transmises à l'indicateur. Ce diagramme utilise 16 Ko de mémoire.



L'illustration suivante essaie incorrectement de remédier à ce problème en convertissant le tableau de nombres aléatoires en double précision en un tableau de nombres aléatoires en simple précision. Cet exemple utilise la même quantité de mémoire que le précédent.



La meilleure solution, représentée dans l'illustration suivante, consiste à convertir le nombre aléatoire en simple précision lorsqu'il est créé, avant de créer un tableau. Ceci permet d'éviter la conversion d'un gros buffer de données d'un type de données à un autre.

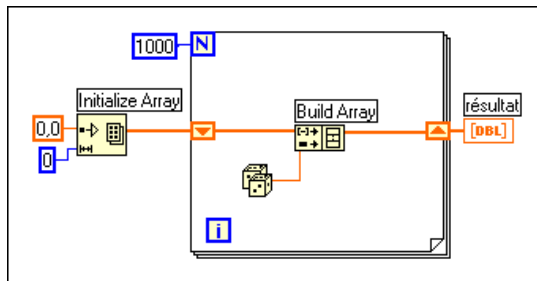


Eviter de redimensionner constamment les données

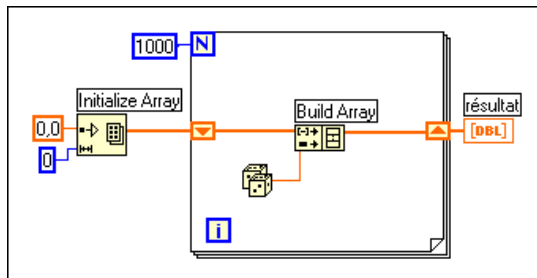
Si la taille d'une sortie est différente de la taille d'une entrée, la sortie ne réutilise pas le buffer de données de l'entrée. C'est le cas pour les fonctions telles que "Construire un tableau", "Concaténer des chaînes de caractères" et "Sous-ensemble d'un tableau", qui augmentent ou diminuent la taille d'un tableau ou d'une chaîne de caractères. Lorsque vous travaillez avec des tableaux et des chaînes de caractères, évitez d'utiliser constamment ces fonctions, sinon votre programme utilise davantage de mémoire de données et s'exécute plus lentement parce qu'il est constamment en train de copier des données.

Exemple 1 : construction de tableaux

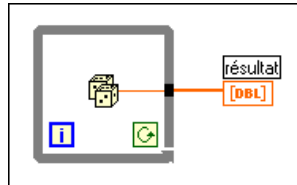
A titre d'exemple, examinez le diagramme suivant, qui est utilisé pour créer un tableau de données. Ce diagramme crée un tableau au sein d'une boucle en appelant constamment la fonction "Construire un tableau" pour concaténer un nouvel élément. Le tableau d'entrée n'est pas réutilisé par "Construire un tableau". A la place, le VI redimensionne continuellement le buffer de sortie pour faire de la place au nouveau tableau, et il copie les données de l'ancien tableau vers le nouveau. La vitesse d'exécution qui en résulte est très lente, en particulier si la boucle est exécutée un grand nombre de fois.



Si vous voulez ajouter une valeur au tableau à chaque itération de la boucle, vous pouvez obtenir les meilleures performances en utilisant une auto-indexation sur la borne de la boucle. Avec les boucles For, le VI peut prédéterminer la taille du tableau (d'après la valeur transmise sur N) et redimensionner le buffer une seule fois.



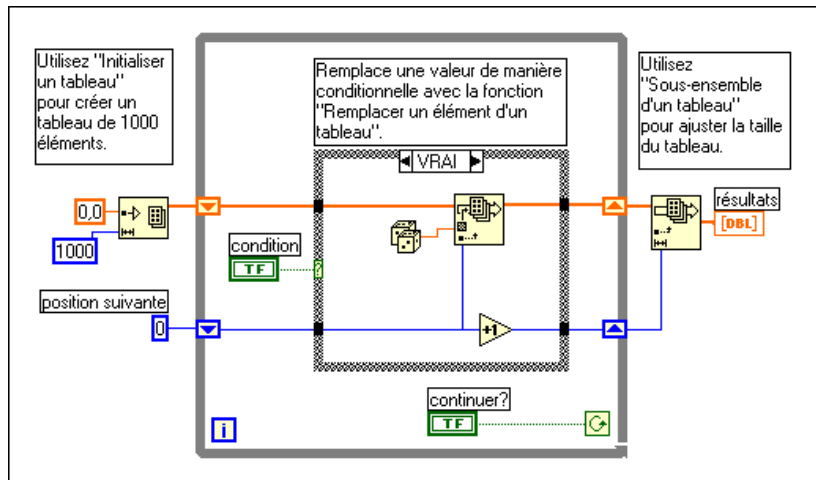
Avec les boucles While, l’auto-indexation ne peut pas être aussi efficace, étant donné que la taille finale du tableau n’est pas connue. Cependant, l’auto-indexation de la boucle While évite de redimensionner le tableau de sortie à chaque itération en augmentant la taille du tableau de sortie par incréments importants. Lorsque la boucle est terminée, le tableau de sortie est redimensionné à la taille correcte. Les performances de l’auto-indexation d’une boucle While sont pratiquement identiques à celles de l’auto-indexation d’une boucle For.



L’auto-indexation suppose que vous ajoutez une valeur au tableau résultant à chaque itération de la boucle. Si vous devez ajouter conditionnellement des valeurs à un tableau, mais si vous pouvez déterminer une limite supérieure de la taille du tableau, vous pouvez envisager de pré-allouer le tableau, puis d’utiliser “Remplacer un élément d’un tableau” pour remplir le tableau.

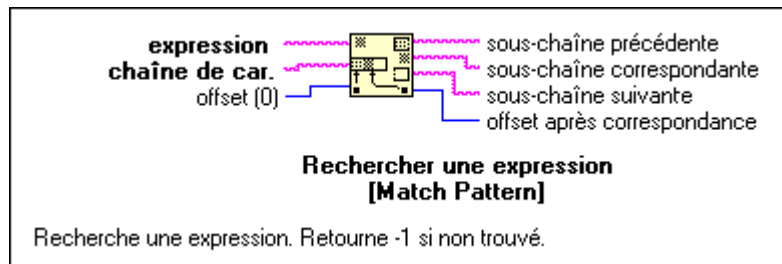
Lorsque vous avez fini de remplir le tableau, vous pouvez redimensionner le tableau à la taille correcte. Le tableau est créé une seule fois, et “Remplacer un élément d’un tableau” peut réutiliser le buffer d’entrée en tant que buffer de sortie. Les performances de cette approche sont très similaires aux performances des boucles utilisant une auto-indexation. Si vous utilisez cette technique, veillez à ce que le tableau dans lequel vous remplacez les valeurs soit suffisamment grand pour contenir les données résultantes, parce que “Remplacer un élément d’un tableau” ne redimensionne pas les tableaux pour vous.

Un exemple de ce processus est représenté dans l'illustration suivante :



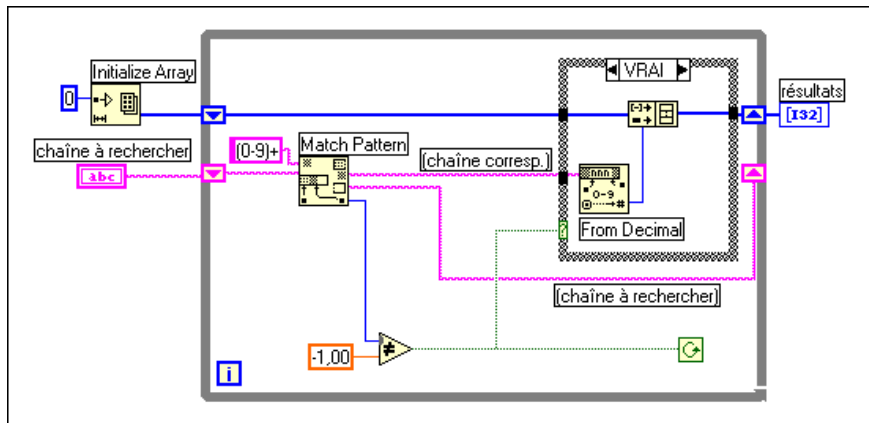
Exemple 2 : recherche dans les chaînes de caractères

Vous pouvez utiliser la fonction “Rechercher une expression” pour chercher une expression dans une chaîne de caractères. Suivant la manière dont vous l'utilisez, vous pouvez ralentir les performances en créant inutilement des buffers de données de chaîne de caractères. L'illustration suivante montre la fenêtre d'aide de la fonction “Rechercher une expression”.



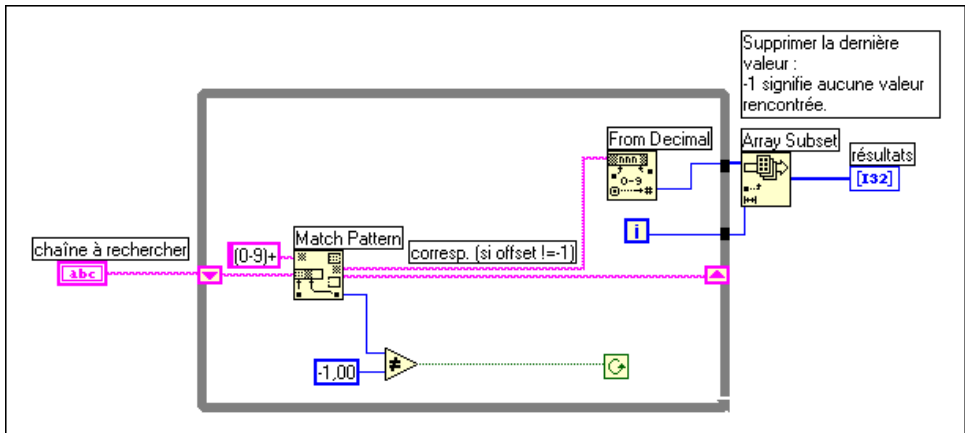
En supposant que vous voulez rechercher un entier dans une chaîne de caractères, vous pouvez entrer $[0-9]^+$ en tant qu'expression courante pour cette fonction. Pour créer un tableau de tous les entiers dans une chaîne de caractères, utilisez une boucle et continuez à appeler "Rechercher une expression" jusqu'à ce que l'offset renvoyé ait une valeur de -1 .

Le diagramme suivant illustre une méthode de balayage pour détecter toutes les occurrences d'entiers dans une chaîne de caractères. Il crée initialement un tableau vide, puis, à chaque itération de la boucle, il examine le reste de la chaîne de caractères à la recherche de l'expression numérique. Si l'expression est trouvée (l'offset n'est pas égal à -1), ce diagramme utilise "Construire un tableau" pour ajouter le nombre à un tableau de nombres répondant au critère de recherche. Lorsqu'il ne reste plus de valeur dans la chaîne de caractères, "Rechercher une expression" renvoie une valeur de -1 et l'exécution de ce diagramme se poursuit.

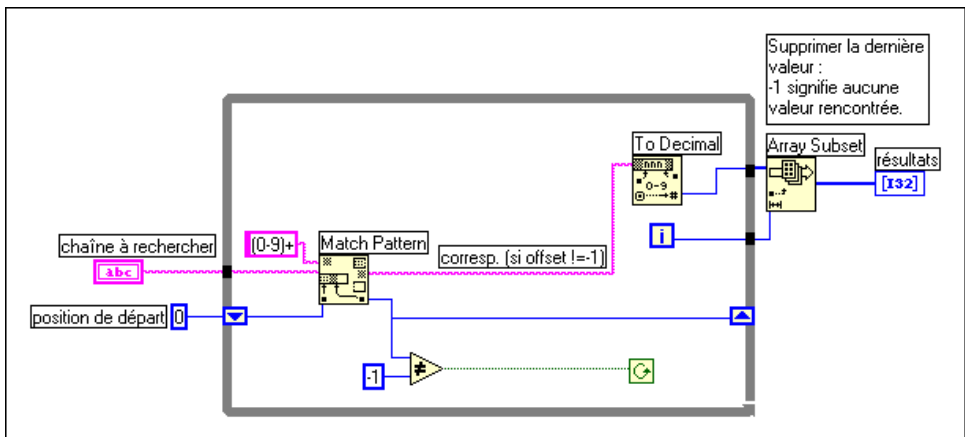


Le problème avec ce diagramme est qu'il utilise "Construire un tableau" dans la boucle pour concaténer la nouvelle valeur aux valeurs précédentes. Au lieu de cela, vous pouvez utiliser une auto-indexation pour accumuler des valeurs sur la borne de la boucle. Notez que vous obtenez une valeur supplémentaire injustifiée dans le tableau, provenant de la dernière itération de la boucle, au cours de laquelle "Rechercher une expression" ne trouve pas de valeur répondant au critère. Une solution consiste à utiliser un sous-

ensemble de tableau pour supprimer cette valeur supplémentaire. Ceci est représenté dans l'illustration suivante :



L'autre problème avec ce diagramme est que vous créez une copie superflue du reste de la chaîne de caractères à chaque itération de la boucle. "Rechercher une expression" a une entrée que vous pouvez utiliser pour indiquer un endroit à partir duquel vous voulez entamer la recherche. Si vous vous souvenez de l'offset de l'itération précédente, vous pouvez utiliser ce nombre pour indiquer où il faut commencer à chercher lors de l'itération suivante. Cette technique est représentée dans l'illustration suivante :

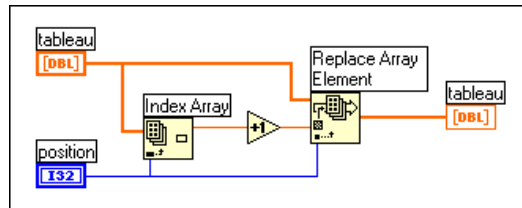


Développement de structures de données efficaces

Un des arguments développés dans la section précédente est que les structures de données hiérarchiques, telles que les tableaux de clusters contenant de grands tableaux ou chaînes de données ou les clusters contenant de grands tableaux ou chaînes de données, ne peuvent pas être manipulées efficacement. Cette section explique pourquoi il en est ainsi et décrit des stratégies pour choisir des types de données plus performants.

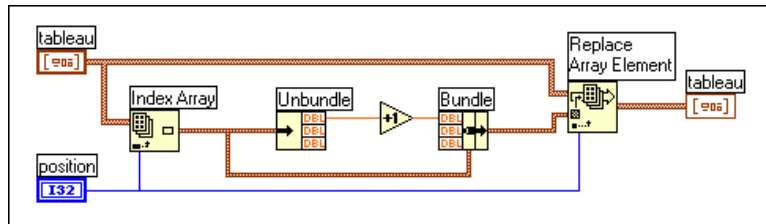
Le problème avec les structures de données compliquées est qu'il est difficile de lire et de modifier des éléments au sein d'une structure de données sans générer de copie des éléments auxquels vous accédez. Si la taille de ces éléments est importante, comme lorsque l'élément lui-même est un tableau ou une chaîne de caractères, ces copies supplémentaires consomment plus de mémoire et de temps pour copier la mémoire.

Vous pouvez généralement manipuler très efficacement les types de données scalaires. De même, vous pouvez manipuler efficacement les petites chaînes de caractères et les petits tableaux dont les éléments sont des scalaires. Dans le cas d'un tableau de scalaires, le code suivant montre ce qu'il faut faire pour incrémenter une valeur dans un tableau.



Ceci est relativement efficace parce qu'il n'est pas nécessaire de générer des copies supplémentaires de l'ensemble du tableau. De plus, l'élément produit par la fonction "Indexer un tableau" est un scalaire, qui peut être créé et manipulé de façon efficace.

La même chose s'applique à un tableau de clusters, en supposant que le cluster contienne seulement des scalaires. Dans le diagramme suivant, la manipulation des éléments devient un peu plus compliquée, parce que vous devez maintenant utiliser Assembler et Désassembler. Cependant, étant donné que le cluster est probablement petit (les scalaires utilisent très peu de mémoire), le temps machine nécessaire pour accéder aux éléments du cluster et les replacer dans le cluster d'origine est négligeable.



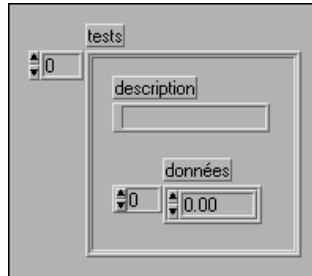
Si vous avez un tableau de clusters dans lequel chaque cluster contient de grands sous-tableaux ou chaînes de caractères, il peut être plus coûteux, en termes de mémoire et de vitesse, de changer la valeur d'éléments du cluster.

Lorsque vous indexez un élément dans le tableau global, une copie de cet élément est créée. Par conséquent, une copie du cluster et du grand sous-tableau ou chaîne de caractères correspondant est réalisée. Étant donné que les chaînes de caractères et tableaux sont de taille variable, le processus de copie peut exiger des appels d'allocation de mémoire pour créer une chaîne de caractères ou un sous-tableau de la taille appropriée, en plus du temps machine nécessaire à la copie proprement dite des données d'une chaîne de caractères ou d'un sous-tableau. Cela peut être négligeable si vous prévoyez de ne le faire que quelques fois. Cependant, si votre application a besoin d'accéder fréquemment à cette structure de données, la mémoire et les délais supplémentaires peuvent rapidement s'accumuler.

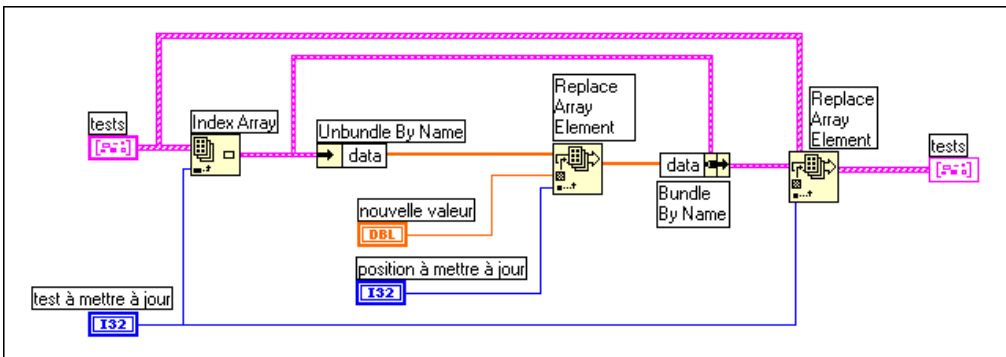
La solution consiste à chercher des représentations alternatives pour vos données. Les trois études de cas suivantes présentent trois applications différentes, ainsi que des suggestions indiquant les meilleures structures de données dans chaque cas.

Etude de cas 1 : éviter les types de données compliqués

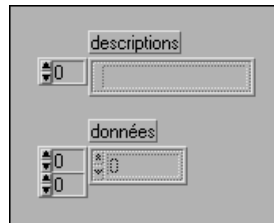
Imaginez une application dans laquelle vous voulez enregistrer les résultats de plusieurs tests. Dans les résultats, vous voulez générer une chaîne de caractères décrivant le test et un tableau de résultats des tests. Un type de données que vous pouvez envisager d'utiliser pour stocker ces informations est représenté dans l'illustration suivante :



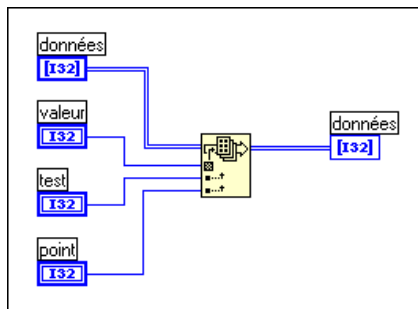
Pour changer un élément dans le tableau, vous devez indexer un élément du tableau global. Vous devez maintenant séparer les éléments du cluster pour atteindre le tableau. Ensuite, vous devez remplacer un élément du tableau et stocker le tableau résultant dans le cluster. Enfin, vous devez stocker le cluster résultant dans le tableau d'origine. Un exemple de cette opération est représenté dans l'illustration suivante :



Chaque niveau de séparation/indexation peut donner lieu à la création d'une copie de ces données. Notez qu'une copie n'est pas systématiquement générée. Copier des données est coûteux, à la fois en termes de temps et en termes de mémoire. La solution consiste à essayer de rendre les structures de données aussi plates que possible. Par exemple, dans cette étude de cas, décomposez la structure de données en deux tableaux. Le premier tableau est le tableau des chaînes de caractères. Le deuxième tableau est un tableau 2D, dans lequel chaque rangée contient les résultats d'un test donné. Ce résultat est représenté dans l'illustration suivante :



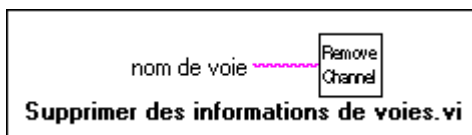
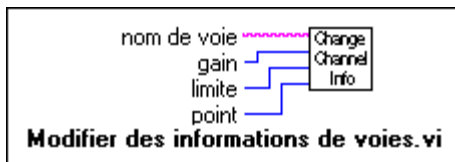
Compte tenu de cette structure de données, vous pouvez remplacer directement un élément du tableau à l'aide de la fonction “Remplacer un élément d’un tableau”, comme cela est représenté dans l'illustration suivante.



Etude de cas 2 : table globale de types de données mélangés

Voici une autre application dans laquelle vous voulez tenir à jour une table d'informations. Dans cette application, vous voulez que les données soient accessibles globalement. Cette table peut contenir des paramètres de configuration d'un instrument, notamment le gain, les limites inférieure et supérieure de tension, et un nom utilisé pour désigner la voie.

Pour rendre les données accessibles dans toute votre application, vous pouvez envisager de créer un ensemble de sous-VIs pour accéder aux données de la table, tels que les sous-VIs suivants, le VI “Modifier des informations de voies” et le VI “Supprimer des informations de voies”.



Les sections suivantes présentent trois approches différentes pour ces VIs.

Approche évidente

Avec cet ensemble de fonctions, il y a plusieurs structures de données à envisager pour la table sous-jacente. D’abord, vous pouvez utiliser une variable globale contenant un tableau de clusters, dans lequel chaque cluster contient le gain, la limite inférieure, la limite supérieure et le nom de la voie.

Comme cela a été décrit dans la section précédente, cette structure de données est difficile à manipuler de façon efficace, parce que vous devez en général effectuer plusieurs niveaux d’indexation et de séparation pour accéder à vos données. En outre, étant donné que la structure de données est un conglomérat de plusieurs segments d’informations, vous ne pouvez pas utiliser la fonction “Rechercher dans un tableau 1D” pour chercher une voie. Vous pouvez utiliser “Rechercher dans un tableau 1D” pour chercher un cluster spécifique dans un tableau de clusters, mais vous ne pouvez pas l’utiliser pour chercher des éléments qui répondent à un élément particulier d’un cluster spécifique.

Approche alternative n° 1

Comme dans l'exemple précédent, choisissez de garder les données dans deux tableaux séparés. L'un contient les noms des voies. L'autre contient les données des voies. L'indice d'un nom de voie donné du tableau de noms est utilisé pour trouver les données de la voie correspondante dans l'autre tableau.

Notez que, puisque le tableau de chaînes de caractères est séparé des données, vous pouvez utiliser la fonction "Rechercher dans un tableau 1D" pour chercher une voie.

En pratique, si vous créez un tableau de 1000 voies à l'aide du VI "Modifier des informations de voies", cette approche est approximativement deux fois plus rapide que la version précédente. Cette amélioration n'est pas très significative parce qu'il y a d'autres facteurs qui affectent les performances.

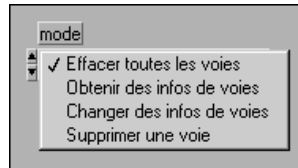
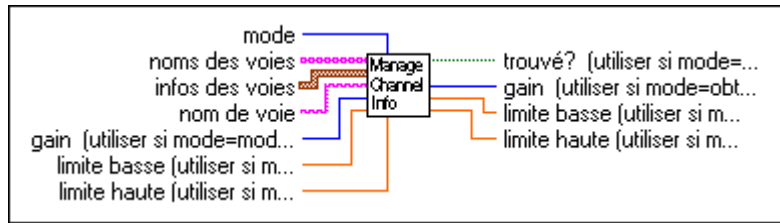
Dans la section *Utilisation de la mémoire par les VIs* de ce chapitre, une remarque met en garde contre l'usage abusif des variables locales et globales. Lorsque vous lisez une variable globale, une copie des données de la variable globale est générée. Par conséquent, une copie des données du tableau complet est générée à chaque fois que vous accédez à un élément. L'approche suivante montre une méthode encore plus efficace pour éviter ce délai supplémentaire.

Approche alternative n° 2

Il existe une méthode pour stocker des données globales, et elle consiste à utiliser un registre à décalage non initialisé. Essentiellement, si vous ne transmettez pas une valeur initiale, un registre à décalage mémorise sa valeur d'un appel à un autre. Si vous êtes un utilisateur LabVIEW et si les registres à décalage non initialisés ne vous sont pas familiers, consultez le chapitre 3, *Boucles et graphes déroulants*, du *Manuel de l'utilisateur LabVIEW* avant de poursuivre la lecture de cette section. Si vous êtes un utilisateur BridgeVIEW, consultez le chapitre 10, *Loops and Charts*, du *BridgeVIEW User Manual* pour des informations similaires.

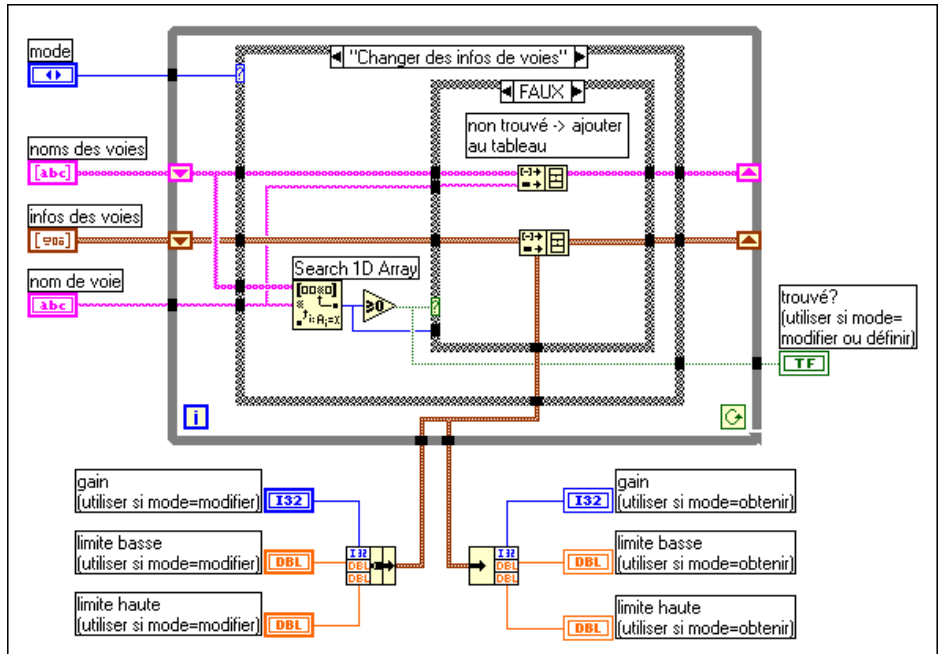
Le compilateur G gère efficacement l'accès aux registres à décalage. La lecture d'une valeur d'un registre à décalage ne génère pas nécessairement une copie des données. En fait, vous pouvez indexer un tableau stocké dans un registre à décalage, et même changer et mettre à jour sa valeur, sans générer de copie supplémentaire du tableau global. Le problème avec un registre à décalage est que seul le VI qui contient le registre à décalage peut accéder aux données de ce dernier. Par contre, le registre à décalage offre l'avantage de la modularité.

Vous pouvez créer un sous-VI unique avec une entrée de mode qui spécifie si vous voulez lire, changer ou supprimer une voie, ou si vous voulez effacer les données pour toutes les voies, comme cela est représenté dans l'illustration suivante :



Le sous-VI contient une boucle While avec deux registres à décalage : une pour les données des voies et l'autre pour les noms des voies. Aucun des deux registres à décalage n'est initialisé. Ensuite, vous placez à l'intérieur de la boucle While une structure Condition connectée à l'entrée de mode. Suivant la valeur du mode, vous pourrez lire et éventuellement changer les données dans le registre à décalage.

Ci-dessous se trouve un schéma d'un sous-VI avec une interface qui gère ces trois modes différents. Seul le code "Modifier des informations de voies" est représenté.

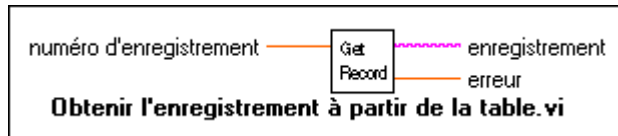
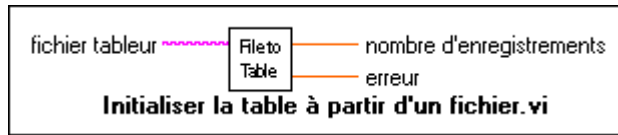


Pour 1000 éléments, cette approche est deux fois plus rapide que l'approche précédente et quatre fois plus rapide que la première approche.

Etude de cas 3 : une table de chaînes de caractères globale statique

L'exemple précédent portait sur une application dans laquelle la table contenait des types de données mélangés, et la table pouvait changer fréquemment. Dans de nombreuses applications, vous avez une table d'informations qui est relativement statique une fois qu'elle est créée. La table peut être lue dans un fichier tableur. Une fois qu'elle est lue et mise en mémoire, vous l'utilisez principalement pour consulter des informations.

Dans ce cas, votre approche peut reposer sur les deux fonctions suivantes, “Initialiser une table à partir d’un fichier” et “Obtenir un enregistrement à partir d’une table”.



Une façon de représenter la table est d’utiliser un tableau de chaînes de caractères à deux dimensions. Notez que le compilateur enregistre chaque chaîne de caractères d’un tableau de chaînes de caractères dans un bloc de mémoire séparé. S’il y a un grand nombre de chaînes de caractères (par exemple, plus de 5000 chaînes de caractères), cela peut représenter une lourde charge sur le gestionnaire de mémoire. Cette charge peut provoquer une baisse notable des performances à mesure que le nombre d’objets augmente.

Une méthode alternative pour stocker une grande table consiste à lire la table comme une chaîne de caractères unique. Ensuite, construisez un tableau séparé contenant les offsets de chaque enregistrement dans la chaîne de caractères. Cela change l’organisation de sorte que, au lieu d’avoir potentiellement des milliers de blocs de mémoire relativement petits, vous avez un gros bloc de mémoire (la chaîne de caractères) et un bloc de mémoire séparé plus petit (le tableau des offsets).

Cette méthode peut être plus compliquée à mettre en œuvre, mais elle peut être beaucoup plus rapide pour les grandes tables.

Portabilité et localisation

Ce chapitre aborde les problèmes concernant le port de VIs d'une plate-forme à une autre et la localisation.

VIs portables et non portables

Les VIs peuvent être portés entre toutes les plates-formes sur lesquelles fonctionne votre application, à condition que les versions de l'application soient les mêmes. Les VIs contenant des CIN ou des fonctions spécifiques à une plate-forme, telles une DDE, ne sont pas portables. Le cas échéant, le VI est porté, mais est invalide.

Le G utilise le même format de fichier sur toutes les plates-formes. Vous pouvez transférer des VIs d'un système à un autre, soit au moyen de disques, soit par l'intermédiaire d'un réseau.

Une fois que le fichier est sur le nouveau système, vous pouvez ouvrir le VI. Votre application détecte que le VI provient d'une autre plate-forme et le recompile pour utiliser les instructions correctes pour le processeur actuel. Si vous transférez les VIs sur un disque formaté pour une plate-forme différente, vous pouvez avoir besoin d'un programme utilitaire, tel que Apple File Exchange sur Macintosh, pour lire le disque.

Vous ne pouvez pas porter les VIs suivants.

- Les VIs distribués dans le répertoire `vi.lib`. Chaque distribution contient son propre `vi.lib`, aussi ne transférez pas les VIs qui se trouvent dans `vi.lib` d'une plate-forme à une autre.
- Les VIs contenant des CIN. Vous obtenez une erreur `code objet non trouvé` si le CIN provient d'une plate-forme différente. Si vous écrivez le code source de votre CIN d'une manière indépendante de la plate-forme, vous pouvez le recompiler sur une autre plate-forme et refaire une édition de liens avec le VI porté.
- Les VIs utilisant le nœud "Appeler une fonction d'une LLB". Le VI peut être porté, mais il est invalide, à moins qu'il ne puisse trouver une bibliothèque du même nom.

- Les VIs de communication spécifiques à une plate-forme, tels que AppleEvents sur Macintosh et DDE sous Windows.
- Les VIs utilitaires “Port d’entrée” (In Port.vi) et “Port de sortie” (Out Port.vi) sous Windows et les VIs utilitaires Peek et Poke pour le Macintosh.

Port d’une plate-forme à une autre

Il y a plusieurs choses que vous pouvez faire pour faciliter le port d’une plate-forme à une autre. Les problèmes de portabilité comprennent des différences au niveau des noms de fichiers, des caractères séparateurs, des résolutions et des fontes, un chevauchement possible des étiquettes et des différences de formats des images.

Un aspect à prendre en considération est le nom des fichiers. Les noms de fichiers sont limités à huit caractères plus une extension optionnelle de trois caractères, généralement .vi sous DOS/Windows 3.x et dans les volumes FAT sous Windows NT. Sur Macintosh, les noms de fichiers peuvent avoir 31 caractères. Sous Windows 95/NT et UNIX, les noms de fichiers peuvent avoir 255 caractères, extension .vi comprise.

Pour éviter les complications, enregistrez les VIs avec des noms courts ou enregistrez-les dans une bibliothèque de VIs. Une bibliothèque de VIs est un fichier unique qui peut contenir plusieurs VIs. Un nom de bibliothèque doit être conforme aux limites de la plate-forme, mais les VIs qui se trouvent dans les bibliothèques peuvent avoir des noms contenant jusqu’à 255 caractères, quelle que soit la plate-forme. Par conséquent, les bibliothèques de VIs constituent le format le plus pratique pour transférer des VIs, étant donné que les bibliothèques éliminent en grande partie la dépendance au système de fichiers. Pour plus d’informations sur la création des bibliothèques de VIs, consultez la section *Enregistrement des VIs* du chapitre 2, *Edition des VIs*.

Un autre problème concernant les noms de fichiers est qu’ils respectent la casse sous UNIX et qu’ils ignorent la casse ailleurs. Si vous faites référence à un VI en le désignant par son nom, par exemple, vous devez veiller à utiliser des majuscules exactement de la même façon à chaque fois.

Différences entre les caractères séparateurs

Si vous développez un VI destiné à être utilisé sur plusieurs plates-formes, n’utilisez pas de caractère séparateur de chemin spécifique à une plate-forme [\ , / , ou :] dans vos noms de fichiers. Évitez les caractères spéciaux dans vos noms de fichiers parce qu’ils peuvent être interprétés

différemment par d'autres systèmes de fichiers. Par exemple, sous UNIX, les fichiers cachés commencent par un point.

Différences de résolution et de fonte

Un autre problème de portabilité concerne les différences au niveau de la résolution de l'écran et des fontes. Les fontes peuvent varier d'une plate-forme à une autre, aussi après avoir porté un VI, il peut être nécessaire de choisir de nouvelles fontes pour obtenir un affichage attrayant. Lorsque vous concevez des VIs, n'oubliez pas qu'il y a trois fontes pour lesquelles la correspondance est optimale d'une plate-forme à une autre : la fonte de l'application, la fonte du système et la fonte des boîtes de dialogue.

Ces fontes prédéfinies et les fontes réelles par lesquelles elles sont représentées sont les suivantes.

- La fonte de l'application est la fonte par défaut. Elle est utilisée dans la palette **Commandes**, dans la palette **Fonctions** et pour les nouvelles commandes.
 - **(Windows)** La version américaine de Windows utilise généralement la fonte Arial. La taille dépend des paramètres du driver vidéo, étant donné que vous pouvez fréquemment installer des drivers vidéo à résolution plus élevée pour utiliser des fontes de grande taille ou de petite taille. Dans la version japonaise de Windows, l'application utilise la fonte que Windows utilise pour les noms de fichiers dans le gestionnaire de programmes.
 - **(Macintosh)** L'application utilise la fonte qui est utilisée dans le Finder pour les noms de fichiers en tant que fonte de l'application. Par exemple, sur le système Macintosh américain, l'application utilise Geneva, alors que sur le système Macintosh japonais, l'application utilise Osaka.
 - **(UNIX)** L'application utilise Helvetica par défaut.
- La fonte système est la fonte utilisée pour les menus.
 - **(Windows)** L'application utilise Helvetica, avec une taille qui dépend du driver vidéo.
 - **(Macintosh)** L'application utilise normalement Chicago dans le logiciel du système américain, Osaka dans le logiciel du système japonais, et ainsi de suite.
 - **(UNIX)** L'application utilise normalement Helvetica pour cette fonte.

- La fonte des boîtes de dialogue est la fonte que le G utilise pour le texte dans les boîtes de dialogue.
 - **(Windows)** Dans la version américaine de Windows, cette fonte est une version à caractères gras de la fonte de l'application. Dans la version japonaise de Windows, cette fonte est la même que la fonte système.
 - **(Macintosh)** L'application utilise la fonte qui est utilisée en tant que fonte système.
 - **(UNIX)** L'application utilise normalement Helvetica pour cette fonte.

Lorsque vous transférez un VI contenant l'une de ces fontes vers une autre plate-forme, votre application veille à ce que la fonte soit représentée par une fonte similaire sur la nouvelle plate-forme.

Si vous n'utilisez pas les fontes prédéfinies mais sélectionnez à la place une fonte spécifique telle que Geneva ou New York, la fonte peut changer de taille sur la nouvelle plate-forme en raison des différences entre les fontes disponibles et des différences au niveau de la résolution de l'affichage. Si vous sélectionnez Geneva ou New York sur Macintosh, l'application ne peut pas la reproduire sur Sun ou HP-UX, et l'application utilise la fonte appelée *fixed*. Si vous transférez sur Windows un VI contenant des fontes non reconnues, il est possible qu'il n'y ait pas une bonne correspondance avec une nouvelle fonte.

Si vous utilisez une fonte prédéfinie sur une section de texte, National Instruments recommande que vous ne changiez pas la taille de ce texte. Si vous changez la taille de la fonte pour lui donner une valeur autre que la valeur par défaut, et si vous transférez ensuite le VI vers une autre plate-forme, l'application essaie de faire correspondre la fonte avec la nouvelle taille, ce qui n'est peut-être pas approprié pour la résolution donnée de l'écran. Par exemple, une fonte d'application de taille 10 (un pixel de plus que la taille par défaut) donne de bons résultats sur le Macintosh (Geneva 10) mais paraît minuscule sous Windows avec un driver vidéo à haute résolution, où elle a trois pixels de moins que la taille par défaut.

Chevauchement d'étiquettes

Lorsque vous transférez un VI vers une nouvelle plate-forme, les commandes et les étiquettes peuvent changer de taille, selon que les fontes sont plus petites ou plus grandes. Le G essaie d'éviter que les étiquettes chevauchent leur propriétaire en les écartant de la commande propriétaire. En outre, chaque étiquette ou constante a un attribut par défaut appelé **Ajuster à la taille du texte**. Lorsque vous créez une étiquette ou une

constante, cet attribut est actif, donc les limites de l'objet se déplacent autant qu'il le faut pour afficher la totalité du texte qui y est inscrit.

Si vous redimensionnez manuellement l'objet, l'application désactive cet attribut (l'élément du menu local n'est plus coché). Une fois l'option **Ajuster à la taille du texte** désactivée, les limites de l'objet restent constantes, et l'application coupe le texte inscrit si cela est nécessaire. Si vous ne voulez pas que l'application coupe du texte lorsque vous passez d'un système à un autre ou d'une plate-forme à une autre, gardez cet attribut actif pour les étiquettes et les constantes.

La plupart des moniteurs Sun et HP sont beaucoup plus grands et ont une résolution plus élevée que les moniteurs PC et Macintosh. Si vous utilisez un Sun ou un HP-UX, ne créez pas de faces-avant très grandes si vous voulez les porter facilement.

Pour garantir les meilleurs résultats possibles, évitez les commandes qui se chevauchent et laissez de la place. Si une étiquette chevauche, même partiellement, un autre objet et si la fonte est agrandie, elle peut finir par recouvrir complètement la commande.

Différences entre les images

Le type d'image le plus élémentaire est une image bitmap, constituée d'une série de valeurs définissant la couleur de chaque pixel de l'image. Les images plus complexes peuvent contenir diverses commandes qui sont exécutées à chaque fois que l'image est dessinée. Les images contenant des commandes de dessin sont créées par des programmes de dessin ou dans la section peinture d'une application graphique.

Les images bitmap constituent un format d'enregistrement commun des images sur toutes les plates-formes. Si vous utilisez des images au format bitmap sur vos faces-avant, les images ont généralement la même apparence lorsque vous chargez vos VIs sur une autre plate-forme. Cependant, les images contenant des commandes de dessin peuvent contenir des commandes qui ne sont pas reconnues sur d'autres plates-formes, comme par exemple la découpe et le remplissage par un motif. Ces images pourront avoir une apparence bizarre sur d'autres plates-formes. Vérifiez à quoi ressemblent vos VIs sur une autre plate-forme sur laquelle ils risquent d'être utilisés.

Vous pouvez toujours utiliser un programme de dessin ou la section dessin d'une application graphique pour créer vos images. Mais pour les rendre plus facilement portables, collez l'image finale dans un programme de

peinture ou dans la section peinture d'une application graphique avant d'importer l'image.

(Windows 95/NT et Macintosh) Avec certaines applications sous Windows 95/NT et de nombreuses applications graphiques sur Macintosh, vous pouvez couper ou copier une image de forme non rectangulaire. Sur Macintosh, par exemple, vous pouvez utiliser un outil Lasso comme celui qui est représenté à gauche pour sélectionner le contour d'un cercle, d'un triangle ou d'une forme plus compliquée comme une note de musique. D'autres plates-formes pourront dessiner ces formes irrégulières sur un fond blanc rectangulaire. Sous Windows 95/NT, utilisez des applications qui supportent des fichiers au format "Enhanced Metafile" si vous voulez des images avec des formes irrégulières et des images dont l'échelle peut être facilement modifiée.



Localisation des VIs

Vous pouvez localiser les chaînes de caractères de la face-avant en les exportant vers un fichier texte muni de tags, en traduisant le fichier texte et en important à nouveau ce fichier dans votre face-avant LabVIEW. Consultez le chapitre 5, *Impression et documentation des VIs*, pour plus d'informations. Lorsque vous traduisez ce fichier texte muni de tags, vous pouvez traduire le titre de la fenêtre du VI, comme cela est représenté dans l'illustration suivante, ou vous pouvez changer le titre de la fenêtre de manière interactive en utilisant les options de configuration du VI. Consultez la section *Edition du titre de la fenêtre du VI*, plus loin dans ce chapitre, et la section *Options des fenêtres* du chapitre 6, *Configuration des VIs et sous-VIs*, pour plus d'informations.



Figure 29-1. Exemple de localisation de VI

Tous les objets de la face-avant peuvent comprendre des étiquettes de sous-titres. Les étiquettes des noms ne peuvent pas être traduites parce que le moteur du langage G utilise cette étiquette pour identifier l'objet. Vous ne pouvez pas changer les étiquettes des noms sans affecter le diagramme, mais vous pouvez traduire les étiquettes des sous-titres et masquer l'étiquette du nom. Pour plus d'informations, consultez la section *L'environnement graphique* du chapitre 2, *Edition des VIs*.

Outre la traduction des chaînes de caractères de la face-avant, vous pouvez utiliser des séparateurs décimaux localisés lorsque vous convertissez des

nombres en chaînes de caractères. Pour plus d'informations, consultez la section *Point et virgule comme séparateurs décimaux*, plus loin dans ce chapitre. “Formater la chaîne de caractères de la date/heure” affiche la date et l'heure conformément à vos spécifications.

Importer et exporter les chaînes de caractères des VIs

L'outil d'exportation/importation des chaînes de caractères des VIs écrit toutes les chaînes de caractères localisables contenues dans la face-avant d'un VI dans un fichier texte muni de tags, qui est appelé “fichier de chaînes de caractères du VI”. Vous devez sélectionner **Projet»Exporter les chaînes de caractères du VI** ou **Projet»Importer les chaînes de caractères du VI**, puis vous devez sélectionner un fichier texte pour l'exportation ou l'importation à partir de la boîte de dialogue de Fichier. Le logiciel n'importe par un fichier de chaînes de caractères d'un VI si le fichier contient des tags inconnus ou s'il manque un tag.

Vous pouvez localiser les chaînes de caractères suivantes.

- titres des fenêtres de VI et descriptions
- étiquettes de sous-titre des objets et descriptions
- étiquettes libres
- valeurs par défaut (chaînes de caractères, tables, chemins et tableaux)
- données privées (noms des éléments des boîtes-listes, en-têtes de rangées et colonnes des tables, noms des tracés du graphe et noms des curseurs du graphe).

L'importation ou l'exportation de chaînes de caractères crée également un fichier d'enregistrement contenant une liste de toutes les erreurs ayant eu lieu au cours de l'opération.

Syntaxe du fichier de chaînes de caractères du VI

Le format du fichier de chaînes de caractères du VI ressemble beaucoup au format d'un fichier HTML. Chaque élément est repéré par un tag de début et un tag de fin. Un tag de début commence par < et se termine par >, et un tag de fin commence par </ et se termine par >. Les espaces blancs sont ignorés, sauf lorsqu'ils sont dans du texte. Etant donné que le caractère < indique le début d'un tag, << est utilisé pour le caractère “inférieur à” dans du texte et, de même, >> est utilisé pour le caractère “supérieur à”. Le caractère “doubles guillemets” est remplacé par " ". De plus, les caractères de fin de ligne sont représentés par <CR>, <CRLF>, ou <LF>, ceux-ci étant traités respectivement comme un retour chariot, un retour chariot suivi d'un retour à la ligne et un retour à la ligne. Ce format est destiné à être lu par

une machine, donc ne vous inquiétez pas si vous le trouvez difficile à lire. Si des tags sont modifiés ou supprimés, des erreurs surviennent lorsque le fichier est importé dans le logiciel.

Le tableau 29-1 dresse la liste des types de tags des VIs et de leur syntaxe.

Tableau 29-1. Description des tags des VIs

Type de tag de VI	Syntaxe du tag de VI
[fichier des chaînes de caractères du VI]	<VI [attributs du VI] > [infos sur le VI] </VI>
[attributs du VI]	syntaxVersion=1 lvVersion=nnn revision=nnn name="texte"
[infos sur le VI]	[titre du VI] [description] [contenu]
[titre du VI]	<TITLE>texte</TITLE> <TITLE><NO_TITLE></TITLE>
[description]	<DESC>texte</DESC>
[contenu]	<CONTENT>[objets]</CONTENT>

Les attributs des VIs sont séparés par un espace, et aucun espace n'est permis entre le nom de l'attribut et le signe égal (=) qui le suit, ni entre le signe égal (=) et la valeur de l'attribut.

Par exemple :

```
<VI syntaxVersion=1 LVversion=4502007 revision=10
name="VI AO-Générer un signal">
```

```
<TITLE>VI AO-Générer un signal</TITLE>
```

```
<DESC>Ce VI génère un signal séquencé à simple buffer
pour la voie de sortie donnée avec la fréquence de mise
à jour donnée.</DESC>
```

```
<CONTENT>
```

```
.....
```

```
</CONTENT>
```

```
</VI>
```

Le tableau 29-2 dresse la liste des tags qui décrivent le contenu de la face-avant, à savoir les étiquettes libres et les étiquettes appartenant à un objet, les étiquettes des sous-titres et les attributs.

Tableau 29-2. Contenu de la face-avant

Type de tag de contenu	Syntaxe du tag de contenu
[contenu]	<CONTENT>[objets]</CONTENT>
[objets]	([commande] [étiquette]) *
[commande]	<CONTROL [attributs de la commande]> [infos sur la commande] </CONTROL>
[étiquette]	<LABEL>[texte de style] </LABEL>
[texte de style]	<STEXT>texte avec des informations sur la fonte </STEXT>

Entre <STEXT> et </STEXT>, vous pouvez taper des spécifications relatives aux fontes. Les informations concernant les fontes sont codées à l'aide du format suivant : . Les attributs des fontes peuvent être listés dans n'importe quel ordre. La spécification de fonte est différente des autres éléments parce qu'elle n'a pas de tag de fin. Par exemple, un sous-titre contenant le texte “*Etiquette caractères gras*” peut être décrit de la façon suivante.

```
<LABEL><STEXT><FONT name="times new roman" size=12
style='I'>Etiquette <FONT style='IB>caractères gras
</STEXT></LABEL>
```

Les fontes peuvent être définies par “predef” pour indiquer l'une des fontes prédéfinies (fonte de l'application, fonte des boîtes de dialogue ou fonte système).

Le tableau 29-3 dresse la liste des tags qui décrivent la [commande].

Tableau 29-3. Tags pour la [commande]

Type de tag de contenu	Syntaxe du tag de contenu
[commande]	<CONTROL [attributs de la commande]> [infos sur la commande] </CONTROL>
[attributs de la commande]	ID=xxx type="Booléen" name="interrupteur"
[infos sur la commande]	[description] [parts] [section privData] [section defData] [contenu]
[parts]	<PARTS> [part]*</PARTS>
[part]	<PART [attributs de la part]> [infos sur la part] </PART>
[attributs de la part]	partID=nnn partOrder=nnn
[infos sur la part]	[commande] [étiquette] [multiLabel]

Ci-dessous se trouve un exemple de description de commande à menu déroulant, dont le sous-titre est “Menu déroulant” et les options sont Charger, Décharger, Ouvrir et Fermer.

```
<CONTROL ID=87 type="Menu déroulant" name="commande à
MENU DEROULANT">

<DESC>commande à menu déroulant</DESC>

<PARTS>

<PART ID=12 order=0 type="Texte du menu déroulant">
<MLABEL><STRINGS><STRING>Charger</STRING>
<STRING>Décharger</STRING><STRING>Ouvrir</STRING>
<STRING>Fermer</STRING></STRINGS></MLABEL></PART>

<PART ID=82 order=0 type="Sous-titre">
<LABEL><STEXT><FONT color=FF0033 size=12>MENU DEROULANT
</STEXT></LABEL></PART>

</PARTS>

</CONTROL>
```


Le tag MLABEL (multilabel) ci-dessus est utilisé pour désigner la chaîne de caractères des options sur une commande à menu déroulant ou les chaînes de caractères sur des boutons booléens, à raison d'une chaîne de caractères pour chacun des quatre états. Ci-dessous figure une description générique de la syntaxe du tag MLABEL.

```
[multiLabel]<MLABEL> [infos mlabel] </MLABEL>
```

```
[infos mlabel][fonte][chaînes de caractères]
```

Le tableau 29-4 dresse la liste des tags qui décrivent les valeurs par défaut pour les chaînes de caractères, les tables, les tableaux et les chemins.

Tableau 29-4. Valeurs par défaut pour les chaînes de caractères

Type de tag de contenu	Syntaxe du tag de contenu
[section defData]	<DEFAULT> [defData] </DEFAULT>
[defData]	[déf chaîne de car] [déf table] [valeurs tableau] [valeurs chemin]
[déf chaîne de car]	[chaîne de car] <SAME_AS_TEXT>
[déf table]	[chaînes de car]
[valeurs tableau]	<ARRAY nElems=n> [valeurs éléments tableau] </ARRAY>
[valeurs éléments tableau]	[valeurs clust] [valeurs chaîne de car] [valeurs non-chaîne de car]
[valeurs chaîne de car]	[chaîne de car]
[valeurs non-chaîne de car]	<NON_STRING>
[valeurs clust]	<CLUSTER nElems=n> [valeurs éléments clust] </CLUSTER>
[valeurs éléments clust]	[valeurs clust] [valeurs chaîne de car] [valeurs non-chaîne de car] [valeurs tableau] [valeurs chemin]
[valeurs chemin]	<PATH type = "absolute"> a<SEP> SYSTEM </PATH>

Pour [valeurs tableau], n [valeurs éléments tableau] doivent suivre le tag <ARRAY>. De même, pour [valeurs clust], il doit y avoir n [valeurs éléments clust].

Pour les valeurs par défaut des commandes de chaîne de caractères, utilisez un tag spécial, <SAME_AS_TEXT>, qui indique que la valeur par défaut de la chaîne de caractères est la même que celle de l'étiquette kStrTextID de la liste de parts de la chaîne de caractères. L'emploi de ce tag élimine le besoin de répéter le même texte pour l'étiquette kStrTextID et la valeur par défaut de la chaîne de caractères.

Pour les valeurs par défaut des commandes de chemin, le tag de début <PATH> peut avoir un attribut qui spécifie le type de chemin. Les valeurs d'attribut possibles sont "absolute", "relative", "not-a-path" et "unc". Les segments de chemin situés entre les tags <PATH> et </PATH> sont délimités par un tag <SEP>. Par exemple, sur une plate-forme Windows, le chemin absolu c:\windows\temp\temp.txt est écrit de la manière suivante.

```
<PATH type="absolute">c<SEP>windows<SEP>temp<SEP>temp.txt</PATH>
```

Le tableau 29-5 dresse la liste des tags qui décrivent des données privées, telles que les noms des éléments des boîtes-listes, les en-têtes de rangées et colonnes des tables, les noms des tracés du graphe et les noms des curseurs du graphe.

Tableau 29-5. Description des tags pour les cellules des tables, les noms des tracés du graphe et les noms des curseurs

Type de tag de contenu	Syntaxe du tag de contenu
[section privData]	<PRIV> [privData] </PRIV>
[privData]	([éléments] [en-tête de col] [en-tête de rangée] [fontes des cellules] [tracés] [curseurs])
[éléments]	<ITEMS> [chaîne de caractères]* </ITEMS>
[en-tête de col]	<COL_HEADER> [chaîne de caractères]* </COL_HEADER>
[en-tête de rangée]	<ROW_HEADER> [chaîne de caractères]* </ROW_HEADER>

Tableau 29-5. Description des tags pour les cellules des tables, les noms des tracés du graphe et les noms des curseurs (Suite)

Type de tag de contenu	Syntaxe du tag de contenu
[fontes des cellules]	<CELL_FONTS> [fonte de cellule]* </CELL_FONTS>
[tracés]	<PLOTS> [chaîne de caractères]* </PLOTS>
[curseurs]	<CURSORS> [chaîne de caractères]* </CURSORS>
[fonte de cellule]	[row# col#][fonte]
[fonte]	

[chaînes de caractères] et [chaîne de caractères] ont le format suivant :

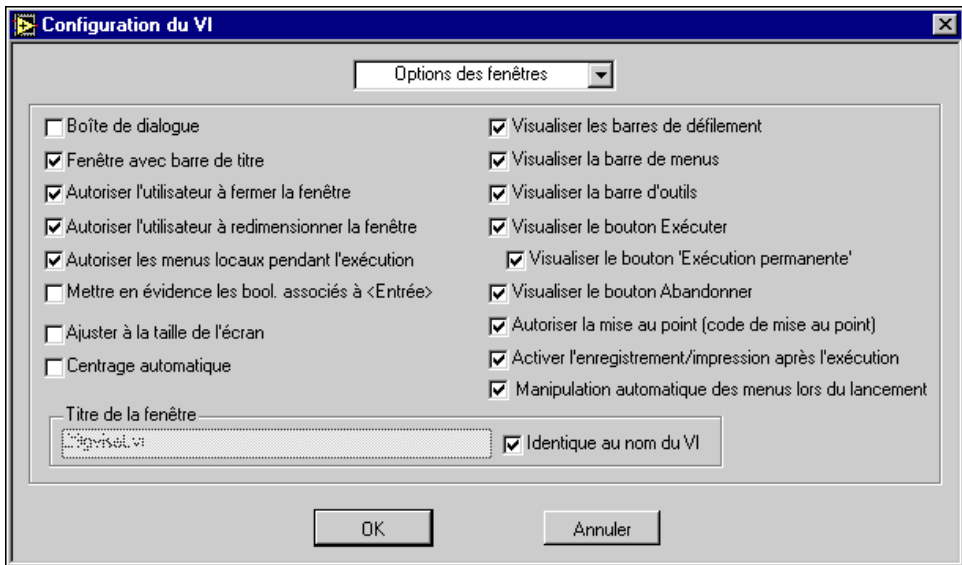
```
[chaînes de caractères]<STRINGS> [chaîne de caractères]*
</STRINGS>
```

```
[chaîne de caractères]<STRING> texte </STRING>
```

Edition du titre de la fenêtre du VI

Vous pouvez personnaliser le titre de la fenêtre du VI pour qu'il soit plus descriptif que le nom de fichier du VI. Cette fonction est importante pour les VIs localisés. Le titre de la fenêtre du VI peut être traduit dans la langue locale, il n'est pas soumis aux contraintes imposées sur les noms par le système de fichiers et il est toujours reconnu par les VIs qui peuvent l'appeler. Pour changer le titre de la fenêtre du VI pendant que vous éditez le VI, sélectionnez **Configuration du VI** et **Options des fenêtres** dans le

menu déroulant principal, comme cela est représenté dans l'illustration suivante.



Pour changer le **Titre de la fenêtre**, désélectionnez **Identique au nom du VI** et tapez le titre désiré pour la fenêtre du VI. Pour changer le titre de la fenêtre d'un VI au moyen d'un programme, consultez le chapitre 21, *VI Serveur*.

Point et virgule comme séparateurs décimaux

Vous pouvez choisir d'utiliser le séparateur décimal du système ou d'utiliser un point, en sélectionnant **Face-avant** dans le menu déroulant qui se trouve dans la boîte de dialogue **Edition**»**Préférences**. Si vous le préférez, vous pouvez imposer l'emploi d'un point comme séparateur décimal lorsque vous convertissez des nombres en chaînes de caractères, et vice versa, à l'aide des fonctions suivantes.

- Au format ingénieur
- Au format fractionnel
- Au format exponentiel
- A partir d'un format exp/fract/ing

Formater la chaîne de caractères de la date/heure

Vous pouvez régler le format d’affichage de la date et de l’heure à l’aide de la fonction **Formater la chaîne de caractères de la date/heure**. Pour plus d’informations sur cette fonction, consultez la *Référence en ligne*, disponible en ligne en sélectionnant **Aide**»**Référence en ligne**.

Port d’une application en langage G à une autre

Tout VI LabVIEW qui contient toujours son diagramme, peut-être converti pour BridgeVIEW, à condition que la version du langage G de LabVIEW soit compatible avec celle de BridgeVIEW. Convertir des VIs de BridgeVIEW à LabVIEW est également possible, mais il y a des VIs pour lesquels la conversion n’est pas réalisable.

De LabVIEW à BridgeVIEW

Le facteur le plus important à noter lorsque vous convertissez des VIs de LabVIEW à BridgeVIEW est la version du langage de programmation en G qui a été utilisée pour créer le VI. Consultez le tableau suivant. Tous les VIs LabVIEW créés avec LabVIEW 3.x à LabVIEW 4.x peuvent être chargés dans n’importe quelle version de BridgeVIEW, à condition que les VIs aient toujours leurs diagrammes. Les VIs créés avec LabVIEW 5.0.x peuvent être chargés uniquement par BridgeVIEW 2.0 ou une version plus récente.

Version LabVIEW	Version G	Version BridgeVIEW	Version G
n’importe laquelle jusqu’à 4.0.1	n’importe laquelle jusqu’à 4.0.1	1.0	4.1
4.1	4.0.2	1.0.1	4.1.1
4.1.1	4.0.3	1.1	4.1.2
5.0	5.0	2.0	5.0

De BridgeVIEW à LabVIEW

Il y a deux facteurs qui déterminent si un VI BridgeVIEW peut être chargé et exécuté par LabVIEW. Premièrement, les versions du langage G doivent être compatibles. Par conséquent, les VIs BridgeVIEW doivent être chargés par LabVIEW 5.0 ou une version ultérieure, étant donné que le G 4.0.x ne peut pas charger les VIs créés par le G 4.1.x ou une version plus récente. Le deuxième facteur qui détermine si un VI peut être converti de BridgeVIEW à LabVIEW est si le VI contient des VIs spécifiques à BridgeVIEW. Un VI BridgeVIEW qui utilise des fonctions spécifiques à BridgeVIEW peut ne pas se charger ou ne pas fonctionner correctement sous LabVIEW. Par exemple, le type de données tag est spécifique à BridgeVIEW. Tout VI qui utilise le type de données tag ou accède au moteur de BridgeVIEW ne fonctionne pas sous LabVIEW.

Si vous envisagez d'utiliser BridgeVIEW pour développer des VIs que vous avez l'intention de convertir à LabVIEW, sélectionnez le jeu de palettes G ou LabVIEW de base. Ce jeu de palettes affiche les mêmes fonctions que le jeu de palettes LabVIEW. Les VIs construits en utilisant seulement les fonctions de ces palettes sont portables entre BridgeVIEW et LabVIEW.

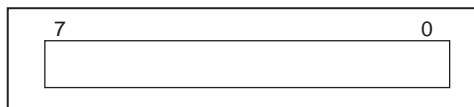
Formats d'enregistrement des données

Cette annexe décrit les formats sous lesquels vous pouvez enregistrer des données. Ces informations sont surtout utiles pour les utilisateurs expérimentés, comme ceux qui doivent utiliser des CIN (code interface nodes) et ceux qui doivent lire et écrire des fichiers utilisés par les fonctions d'E/S sur fichiers. Cette annexe explique comment les données sont enregistrées en mémoire, la relation entre les descripteurs de type et l'enregistrement des données et la méthode dont les données sont aplaties pour l'enregistrement des fichiers sur le disque.

Formats des données pour les commandes et indicateurs de la face-avant

Booléens

Les booléens sont enregistrés comme des valeurs de 8 bits. Si la valeur est nulle, le booléen est FALSE. Toute valeur non nulle représente TRUE.

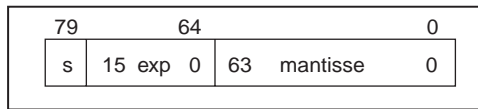


Nombres

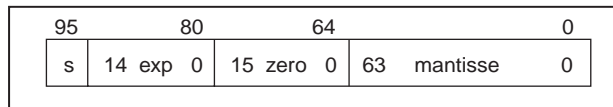
Précision étendue

Lorsque des nombres en précision étendue sont enregistrés sur disque, ils sont conservés dans un format de 128 bits indépendant de la plate-forme, qui est le même que le format en mémoire sous UNIX. En mémoire, la taille et la précision varient suivant la plate-forme.

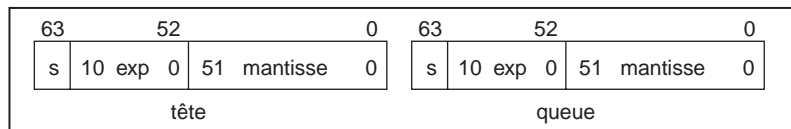
(Windows) Les nombres à virgule flottante en précision étendue ont un format de 80 bits (format en précision étendue 80287).



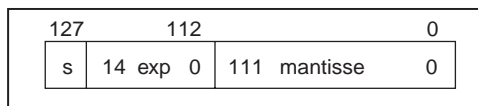
(Macintosh 68K) Les nombres à virgule flottante en précision étendue ont un format de 96 bits (format en précision étendue MC68881-MC68882).



(Power Macintosh) Les nombres à virgule flottante en précision étendue sont représentés comme deux nombres à virgule flottante à double précision combinés, c'est-à-dire, sous le format double-double d'Apple.



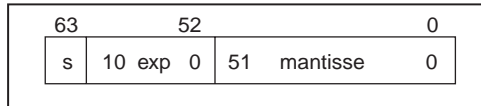
(Sun) Les nombres à virgule flottante en précision étendue ont un format de 128 bits.



(HP-UX) Les nombres à virgule flottante en précision étendue sont représentés comme des nombres à virgule flottante double précision, comme le montre l'illustration suivante.

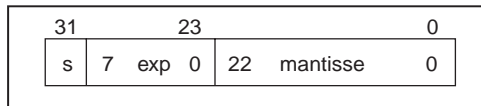
Double précision

Les nombres à virgule flottante double précision sont au format IEEE à 64 bits (format par défaut).



Simple précision

Les nombres à virgule flottante simple précision sont au format simple précision IEEE à 32 bits.



Entier long

Les nombres entiers longs ont un format de 32 bits, signés ou non signés.



Entier mot

Les nombres entiers mots ont un format de 16 bits, signés ou non signés.



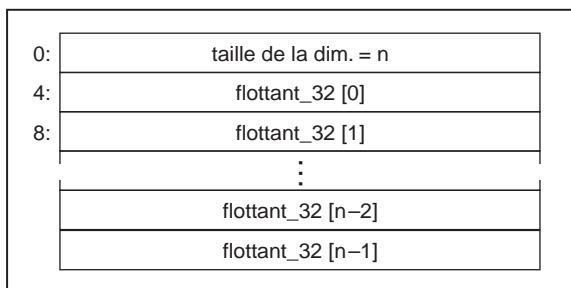
Entier octet

Les nombres entiers octets ont un format de 8 bits, signés ou non signés.

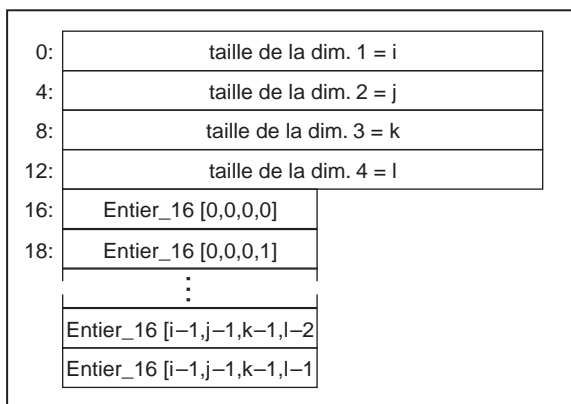


Tableaux

Les tableaux sont enregistrés avec la taille de chaque dimension du tableau en entiers longs, suivis des données. En raison des contraintes d'alignement de certaines plates-formes, la taille de la dimension peut être suivie de quelques octets de remplissage, afin que le premier élément des données soit correctement aligné. Si vous êtes un utilisateur de LabVIEW, consultez la section *Alignment Considerations* du chapitre 2, *CIN Parameter Passing*, du *LabVIEW Code Interface Reference Manual* pour plus d'informations. Ce document est disponible uniquement au format PDF (format document portable) sur le CD ou les disquettes du programme de votre logiciel. L'exemple suivant montre un tableau à une dimension de nombres à virgule flottante simple précision. Les nombres décimaux situés à gauche représentent les offsets en octets des emplacements de mémoire à partir desquels commence le tableau.

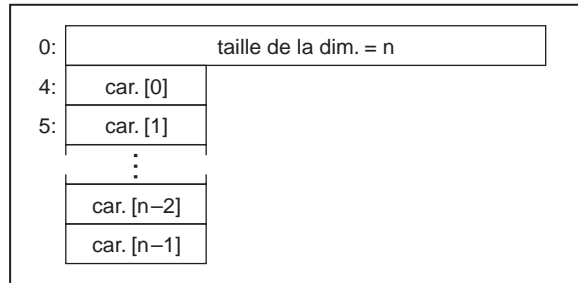


Cette illustration montre un tableau à quatre dimensions d'entiers mots.



Chaînes de caractères

Les chaînes de caractères sont enregistrées comme s'il s'agissait de tableaux à une dimension d'entiers octets (caractères à 8 bits).



Chemins

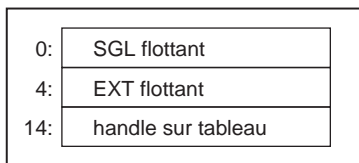
Les chemins sont enregistrés avec le type de chemin et le nombre de composantes du chemin sous forme d'entiers mots, suivis immédiatement des composantes du chemin. Le type de chemin est 0 pour un chemin absolu et 1 pour un chemin relatif. Chaque composante du chemin est une chaîne de caractères Pascal dans laquelle le premier octet est la longueur, en octets, de la chaîne de caractères Pascal (sans compter l'octet de longueur).

Clusters

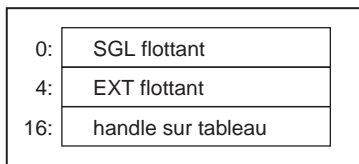
Un cluster enregistre des éléments de divers types de données conformément à l'*ordonnancement du cluster*. L'enregistrement des données scalaires dans le cluster s'effectue de façon directe. Les tableaux, les chaînes de caractères, les handles et les chemins sont enregistrés de façon indirecte. Le cluster enregistre un handle qui pointe vers la zone de mémoire dans laquelle les données sont effectivement enregistrées. En raison des contraintes d'alignement de certaines plates-formes, la taille de la dimension peut être suivie de quelques octets de remplissage, afin que le premier élément des données soit correctement aligné. Si vous êtes un utilisateur de LabVIEW, consultez la section *Alignment Considerations* du chapitre 2, *CIN Parameter Passing*, du *LabVIEW Code Interface Reference Manual* pour plus d'informations. Ce document est disponible uniquement au format PDF (format document portable) sur le CD ou les disquettes du programme de votre logiciel.

Les illustrations suivantes montrent un cluster contenant un nombre à virgule flottante simple précision, un nombre à virgule flottante en précision étendue et un handle pour un tableau à une dimension d'entiers mots non signés, présentés dans cet ordre. Pour plus d'informations, consultez le chapitre 14, *Commandes et indicateurs de tableau et de cluster*.

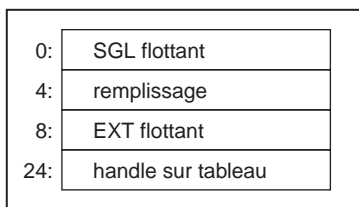
- **(Windows)**



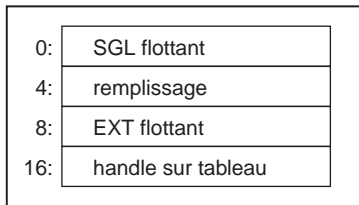
- **(Macintosh)**



- **(Sun)**

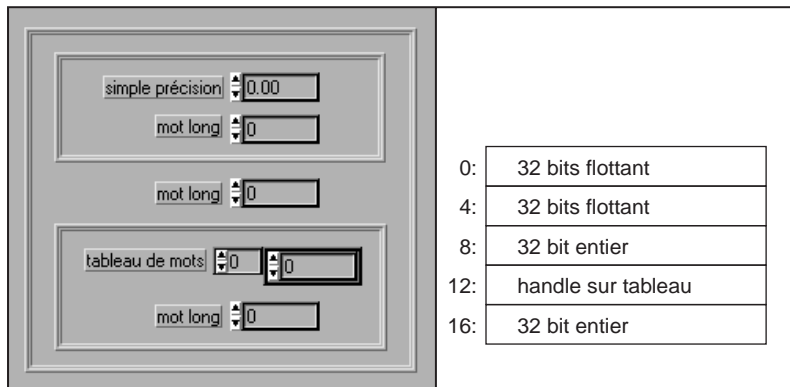


- **(HP-UX)**



Dans l'exemple suivant, les clusters imbriqués ne sont pas enregistrés de façon indirecte. En fait, les données sont enregistrées directement à l'intérieur des clusters imbriqués, comme si les données n'étaient pas

imbriquées dans le sous-cluster. Seuls les tableaux, chaînes de caractères et handles sont enregistrés de façon indirecte.



Descripteurs de type

Chaque fil de liaison et terminal du diagramme sont associés à un type de données. Vous pouvez connaître ce type grâce à une structure en mémoire appelée *descripteur de type*. Ce descripteur est une chaîne d'entiers mots qui peut décrire n'importe quel type de données en G. Les valeurs numériques sont écrites en hexadécimal, sauf mention contraire.

Le format générique d'un descripteur de type est : <longueur> <code de type>.

Certains descripteurs de type comportent, après le code de type, des informations supplémentaires. Les tableaux et les clusters sont des types de données structurés ou regroupés, parce qu'ils comprennent d'autres types. Par exemple, le type cluster contient des informations supplémentaires sur le type de chacun de ses éléments.

Le premier mot (16 bits) de n'importe quel descripteur de type correspond à la longueur, en octets, de ce descripteur de type (y compris le mot de longueur). Le deuxième mot (16 bits) est le code de type. L'octet d'ordre supérieur du code de type est réservé à un usage interne. Lorsque vous comparez deux descripteurs de type pour voir s'ils sont identiques, il est préférable d'ignorer cet octet ; deux descripteurs sont égaux même si les octets d'ordre supérieur des codes de type ne le sont pas.

Le code de type est une version codée des véritables informations de type, telles que nombre à virgule flottante en simple précision ou en précision

étendue, conforme aux indications du tableau suivant. Ces valeurs du code de type pourront changer dans les prochaines versions. Les *xx* dans la colonne du descripteur de type représentent des valeurs réservées. Ignorez-les.

Types de données

Les tableaux suivants donnent les types de données numériques scalaires et non numériques, les codes de type et les descripteurs de type.

Tableau A-1. Types de données numériques scalaires

Type de données	Code de type (nombres en hexadécimal)	Descripteur de type (nombres en hexadécimal)
Entier octet	01	0004 <i>xx</i> 01
Entier mot	02	0004 <i>xx</i> 02
Entier long	03	0004 <i>xx</i> 03
Entier octet non signé	05	0004 <i>xx</i> 05
Entier mot non signé	06	0004 <i>xx</i> 06
Entier long non signé	07	0004 <i>xx</i> 07
Nombre à virgule flottante simple précision	09	0004 <i>xx</i> 09
Nombre à virgule flottante double précision	0A	0004 <i>xx</i> 0A
Nombre à virgule flottante précision étendue	0B	0004 <i>xx</i> 0B
Nombre à virgule flottante complexe simple précision	0C	0004 <i>xx</i> 0C
Nombre à virgule flottante complexe double précision	0D	0004 <i>xx</i> 0D
Nombre à virgule flottante complexe précision étendue	0E	0004 <i>xx</i> 0E

Tableau A-1. Types de données numériques scalaires (Suite)

Type de données	Code de type (nombres en hexadécimal)	Descripteur de type (nombres en hexadécimal)
Entier octet de type énuméré	15	<nn> xx15 <k> <k pstrs>
Entier mot de type énuméré	16	<nn> xx16 <k> <k pstrs>
Entier long de type énuméré	17	<nn> xx17 <k> <k pstrs>
Quantité physique simple précision	19	<nn> xx19 <k> <k base-exp>
Quantité physique double précision	1A	<nn> xx1A <k> <k base-exp>
Quantité physique précision étendue	1B	<nn> xx1B <k> <k base-exp>
Quantité physique complexe simple précision	1C	<nn> xx1C <k> <k base-exp>
Quantité physique complexe double précision	1D	<nn> xx1D <k> <k base-exp>
Quantité physique complexe précision étendue	1E	<nn> xx1E <k> <k base-exp>

Tableau A-2. Types de données non numériques

Type de données	Code de type (nombres en hexadécimal)	Descripteur de type (nombres en hexadécimal)
Booléen	21	0004 xx21
Chaîne de caractères	30	0008 xx30 <len>
Handle	31	0006 xx31 <kind>
Chemin	32	0008 xx32 <len>
Image	33	0008 xx33 <len>

Tableau A-2. Types de données non numériques (Suite)

Type de données	Code de type (nombres en hexadécimal)	Descripteur de type (nombres en hexadécimal)
Tableau	40	<nn> xx40 <k> <k dims> <descripteur de type des éléments>
Cluster	x50	<nn> xx50 <k> <k descripteurs de type des éléments>

La valeur minimale dans le champ de taille d'un descripteur de type est de 4 (comme cela est représenté dans le tableau A-1). Cependant, il est possible d'ajouter un nom (une chaîne de caractères Pascal) à n'importe quel descripteur de type, et dans ce cas la taille est augmentée de la longueur du nom (arrondie à un multiple de 2).

Les types de données des chaînes de caractères, des chemins et des images ont une longueur de 32 bits (identique à la taille de la dimension d'un tableau), bien que la seule valeur couramment codée soit `FFFFFFFF (-1)`, qui indique une taille variable. Tous les chaînes de caractères, chemins et images sont de taille variable. La longueur effective est enregistrée avec les données.

Notez que les types de données des tableaux et des clusters ont chacun leur propre code de type. Ils contiennent également des informations supplémentaires concernant leur nombre de dimensions (pour les tableaux) ou leur nombre d'éléments (pour les clusters), ainsi que des informations concernant les types de données de leurs éléments.

Dans l'exemple suivant d'un entier octet de type énuméré pour les éléments `am, fm, fm stereo`, chaque groupe de caractères représente un mot de 16 bits. L'espace inscrit entre guillemets (" ") représente un espace ASCII.

```
0016 0015 0003 02a m02 fm 09f m" " st er eo
```

0016 indique un total de 22 octets. 0015 indique un entier octet de type énuméré. 0003 indique qu'il y a trois éléments.

Dans l'exemple suivant d'une quantité physique double précision, dont les unités sont exprimées en m/s, chaque groupe représente un mot de 16 bits.

```
000E 001A 0002 0002 FFFF 0003 0001
```

000E indique un total de 14 octets. 0x1A indique qu'il s'agit d'un nombre double en précision avec des unités. 0002 indique deux paires base-exposant. 0002 désigne l'indice de base des secondes. FFFF (-1) est l'exposant des secondes. 0003 désigne l'indice de base des mètres. 0001 est l'exposant des mètres.



Remarque

Toutes les quantités physiques sont enregistrées en termes d'unités de base, quelle que soit l'unité dans laquelle elles sont affichées. Le tableau 9-2, Unités de base, montre les neuf bases qui sont représentées par les indices 0-8 pour les radians-candela.

Tableau

Le code de type pour un tableau est 0x40. Immédiatement après le code de type se trouve un mot contenant le nombre de dimensions du tableau. Ensuite, pour chaque dimension, un entier long contient la taille, en éléments, de cette dimension. Enfin, après chacune des tailles de dimension, apparaît le descripteur de type de l'élément. Le type d'élément peut être n'importe quel type sauf un tableau. La taille de dimension peut être FFFFFFFF (-1) pour n'importe quelle dimension. Cela signifie que la taille de la dimension du tableau est variable. Couramment, tous les tableaux sont de taille variable. La taille de la dimension effective est enregistrée avec les données et est toujours supérieure ou égale à zéro. Ci-dessous se trouve un descripteur de type pour un tableau à une dimension de nombres à virgule flottante double précision :

```
000E 0040 0001 FFFF FFFF 0004 000A
```

000E est la longueur globale du descripteur de type, y compris le descripteur de type des éléments. Le tableau est de taille variable, donc la taille de la dimension est FFFFFFFF. Notez que le descripteur de type des éléments (0004 000A) apparaît exactement de la même manière que pour un scalaire du même type.

Ci-dessous se trouve un exemple d'un descripteur de type pour un tableau à deux dimensions de booléens.

```
0012 0040 0002 FFFF FFFF FFFF FFFF 0004 0021
```

Cluster

Le code de type d'un cluster est 0x50. Immédiatement après le code de type se trouve un mot contenant le nombre d'éléments dans le cluster. Après ce mot se situe le descripteur de type de chaque élément de l'*ordonnancement du cluster*. Par exemple, imaginez un cluster de deux entiers : un entier mot signé et un entier long non signé :

```
000E 0050 0002 0004 0002 0004 0007
```

000E est la longueur du descripteur de type, y compris des descripteurs de type des éléments.

Ci-dessous se trouve un descripteur de type pour un graphe à tracés multiples (les types numériques peuvent varier) :

```
0028 0040 0001 FFFF FFFF... tableau 1D de
001E 0050 0001... cluster à 1 composante de
0018 0040 0001 FFFF FFFF... tableau 1D de
000E 0050 0002... cluster à 2 composantes de
0004 000A... nombre à virgule flottante double précision
0004 0003... entier long
```

Données aplaties

Deux fonctions internes convertissent les données du format en mémoire en une forme mieux appropriée pour les écrire ou les lire dans un fichier.

Etant donné que les chaînes de caractères et les tableaux sont enregistrés sous forme de blocs de handles, les clusters contenant ces types sont *discontigus*. En général, les données sont enregistrées sous la forme d'un *arbre*. Par exemple, un cluster constitué d'un nombre à virgule flottante double précision et d'une chaîne de caractères est enregistré sous forme d'un nombre à virgule flottante de 8 octets, suivi d'un handle de 4 octets pointant vers la chaîne de caractères. Les données de la chaîne de caractères ne sont pas enregistrées à côté du nombre à virgule flottante précision étendue en mémoire. Par conséquent, si vous voulez écrire les données du cluster sur un disque, vous devez aller chercher les données en deux endroits différents. Bien sûr, avec un type de données arbitrairement complexe, les données peuvent être enregistrées en de nombreux endroits différents.

Lorsque des données sont enregistrées dans un fichier de VI ou dans un fichier d'enregistrement de données, ces données sont *aplaties* en une chaîne de caractères unique avant d'être enregistrées. De cette manière,

même les données d'un cluster arbitrairement complexe sont rendues contiguës, au lieu d'être conservées en plusieurs morceaux. Lorsque votre environnement de développement en G charge un fichier de ce type à partir d'un disque, il doit effectuer l'opération inverse : il doit lire une chaîne de caractères unique et la *redresser* pour lui redonner sa forme interne, éventuellement discontiguë.

Les données aplaties sont *normalisées* en une forme standard, afin qu'elles puissent être utilisées sans modification par des VIs fonctionnant sur n'importe quelle plate-forme. Les données numériques sont enregistrées sous forme *big endian* (octet le plus significatif en premier), et les nombres à virgule flottante précision étendue sont enregistrés en tant que quantités de 16 octets à l'aide du format de précision étendue de Sun décrit précédemment dans cette section.



Remarque

Lorsque vous écrivez des données dans un fichier en vue d'une utilisation par une application qui n'a pas été créée en G, vous pouvez avoir intérêt à transformer vos données après les avoir aplaties. De même, lorsque vous lisez des données à partir d'un fichier produit par une application qui n'a pas été créée en G, vous pouvez avoir intérêt à transformer vos données avant de les redresser. Les autres applications pour Windows attendent généralement que les données numériques soient sous forme little endian (octet le moins significatif en premier). Les autres applications pour Windows et Macintosh attendent généralement que les nombres à virgule flottante en précision étendue soient, respectivement, dans les formats de 80 bits et 96 bits décrits plus haut.

Les fonctions du diagramme "Aplatir en chaîne de caractères" et "Redresser à partir d'une chaîne de caractères" (décrite dans l'**Aide en ligne**, disponible en ligne en sélectionnant **Aide**»**Référence en ligne**) vous permettent d'aplatir et de redresser des données exactement de la même manière que le fait votre environnement de développement en G lorsqu'il enregistre et charge des données.

Scalaire

La forme aplatie de n'importe quel type numérique, ainsi que du type booléen, contient seulement les données en format big endian. Par exemple, un entier long avec une valeur de -19 est codé sous la forme FFFF FFED. Un nombre à virgule flottante double précision avec une valeur égale à ¼ est 3FD0 0000 0000 0000. Un TRUE booléen est n'importe quelle valeur non nulle. Un FALSE booléen est 00.

Le format fichier pour les nombres en précision étendue est basé sur le format quadruple précision de SPARC, qui consiste en un exposant de 16 bits suivi d'une mantisse d'une longueur de 112 bits. Sous cette forme, le bit à gauche du point binaire de la mantisse est un bit implicite (supposé toujours égal à 1) qui n'apparaît pas dans la représentation.

Chaînes de caractères, handles et chemins

Etant donné que les chaînes de caractères, les handles et les chemins ont des tailles variables, la forme aplatie est précédée d'un entier long normalisé qui enregistre leur longueur en octets. Pour les chemins, cette longueur est précédée de quatre caractères : PTH0. Par exemple, un type de chaîne de caractères avec une valeur de ABC est aplati sous la forme 0000 0003 4142 43.

Le format aplati est similaire au format que la chaîne de caractères prend en mémoire. Cependant, les handles et les chemins ne sont pas précédés d'une valeur de longueur lorsqu'ils sont chargés en mémoire, donc cette valeur provient de la taille effective des données en mémoire et elle précède les données lorsque celles-ci sont aplaties.

Tableaux

Les données pour un tableau aplati sont précédées par des entiers longs normalisés qui enregistrent la taille, en éléments, de chacune des dimensions des tableaux. La dimension qui varie le plus lentement vient en premier, suivie dans l'ordre par les dimensions suivantes, à variation plus rapide, exactement comme les tailles de dimensions sont enregistrées en mémoire. Les données viennent immédiatement après ces tailles de dimensions, dans l'ordre selon lequel elles sont enregistrées en mémoire. De plus, ces données sont aplaties si nécessaire. Ci-dessous se trouve un exemple de tableau à deux dimensions contenant six entiers de 8 bits.

{ {1, 2, 3}, {4, 5, 6} } est enregistré sous la forme 0000 0002 0000 0003 0102 0304 0506.

Ci-dessous se trouve un exemple d'un tableau à une dimension aplati et contenant des variables booléennes :

{T, F, T, T} est enregistré sous la forme 0000 0004 0100 0101. (01 est la valeur préférée pour TRUE.)

Clusters

Un cluster aplati représente la concaténation (dans l'ordre du cluster) des données aplaties de ses éléments. Donc, par exemple, un cluster aplati constitué d'un entier mot d'une valeur de 4 (décimal) et d'un entier long d'une valeur de 12 s'écrit 0004 0000 000C.

Un cluster aplati constitué d'une chaîne de caractères ABC et d'un entier mot d'une valeur de 4 s'écrit 0000 0003 4142 4300 04.

Un cluster aplati constitué d'un entier mot d'une valeur de 7, d'un cluster contenant un entier mot d'une valeur de 8 et d'un entier mot d'une valeur de 9 s'écrit 0007 0008 0009.

Pour redresser ces données, vous pouvez utiliser simplement le processus inverse. La forme aplatie d'une donnée ne contient *pas* de code de type de données ; le descripteur de type est nécessaire pour cela. La fonction "Redresser à partir d'une chaîne de caractères" exige que vous câbliez un type de données en entrée, afin que la fonction puisse décoder correctement la chaîne de caractères.

Questions courantes concernant le G

Cette annexe apporte des réponses à certaines des questions couramment posées par les utilisateurs du G.

Graphes et graphes déroulants

Que dois-je faire pour envoyer des données vers un graphe ou un graphe déroulant ?

Ouvrez la fenêtre d'aide et faites passer l'outil Bobine devant un terminal de graphe du diagramme pour faire apparaître une brève description de la méthode à utiliser pour câbler des graphes de base. Il y a d'excellents exemples sur les graphes et les graphes déroulants dans le répertoire `examples\general\graphs`.

Quelle est la différence fondamentale entre les graphes et les graphes déroulants ?

Les graphes et les graphes déroulants diffèrent dans la manière dont ils affichent et mettent à jour les données. Les VIs qui utilisent des graphes recueillent généralement les données dans un tableau avant de les afficher sur le graphe, à la manière d'une feuille de calcul qui enregistre d'abord les données avant de générer un tracé. À l'inverse, un graphe déroulant ajoute de nouveaux points à ceux qui sont déjà affichés. De cette manière, vous pouvez voir la valeur ou la mesure actuelle dans le contexte des données préalablement acquises. La longueur du buffer de l'historique du graphe déroulant peut être ajustée au moyen d'une option locale sur le graphe déroulant. Bien sûr, il est également possible d'instaurer un buffer d'historique avec les indicateurs d'un graphe ; cependant, cela doit être fait dans le diagramme. Ces fonctions sont déjà intégrées dans le graphe déroulant.

Que dois-je faire pour inverser les échelles sur mon graphe ou mon graphe déroulant ?

Pour inverser les échelles pour l'axe des x ou l'axe des y , utilisez l'attribut `X Flipped` ou `Y Flipped`. Si l'attribut `X Flipped` est réglé sur `TRUE`, alors le minimum de l'échelle de l'axe des x se situe à droite et le maximum se situe à gauche. De même, si l'attribut `Y Flipped` est réglé sur `TRUE`, le minimum de l'échelle de l'axe des y se situe en haut et le maximum se situe en bas.

Que dois-je faire pour qu'un graphe ou un graphe déroulant affiche des informations dans un format de temps ?

Les graphes et graphes déroulants peuvent afficher des informations numériques au format temps relatif ou temps absolu. Utilisez l'option locale **Echelle des X»Formatage...** ou **Echelle des Y»Formatage...** pour régler le format de l'échelle de l'axe des x ou de l'axe des y sur **Temps relatif**.

Que dois-je faire pour que l'axe des X d'un graphe déroulant affiche le temps réel ?

Consultez l'exemple de VI appelé VI "Graphe déroulant en temps réel" (`Real-Time Chart.vi`), qui se trouve dans la bibliothèque `examples\general\graphs\charts.llb`. Il sépare le cluster date/heure de la fonction "Secondes en date/heure" en heures, minutes et secondes. Il convertit ensuite ces chiffres en un nombre de secondes écoulées depuis minuit. Enfin, ce nombre est utilisé comme entrée pour l'attribut `Xo` et `delta X` d'un graphe déroulant dont le format de l'échelle des x est réglé sur **Temps relatif**. Ou bien, vous pouvez utiliser l'option **Temps absolu** dans **Format & Précision**.

Comment puis-je effacer un graphe déroulant par programme ?

Envoyez un tableau vide vers l'attribut `History Data`. Le type de données de ce tableau vide est le même que le type de données envoyé vers le graphe déroulant. Un excellent exemple illustrant cette technique se trouve dans le VI "Comment effacer les graphes déroulants et les graphes" (`examples\general\graphs\charts.llb\How to Clear Charts & Graphs.vi`).

Que dois-je faire pour éviter que le graphe clignote à chaque fois qu'il est mis à jour ?

Utilisez l'option **Rafraîchissement progressif** pour que les graphes arrêtent de clignoter. Cette option est située dans le sous-menu **Opérations sur les données** du menu local du graphe. **Rafraîchissement progressif** redessine le graphe dans un buffer en dehors de l'écran avant d'afficher l'image. Bien qu'elle donne lieu à une mise à jour plus progressive des indicateurs, cette option est généralement plus lente et nécessite davantage de mémoire.

L'option **Rafraîchissement progressif** peut être activée globalement au moyen de la boîte de dialogue **Edition»Préférences»Face-avant**. Elle peut être activée ou désactivée pour des graphes individuels à partir de leurs menus locaux.

Comment puis-je mettre à jour un graphe sans l'effacer ?

Tous les graphes (forme d'onde, *XY* et intensité) sont systématiquement effacés avant d'écrire de nouvelles données ; cependant, il y a une procédure simple qui permet de conserver les données préalablement écrites sur le graphe et d'y ajouter de nouvelles données à chaque écriture. Le VI "Séparer les valeurs d'un tableau" (exemples\general\arrays.llb\Separate Array Values.vi) démontre la technique consistant à utiliser la fonction "Construire un tableau" pour ajouter de nouvelles valeurs à un tableau. Si vous êtes un utilisateur de LabVIEW, consultez le chapitre 5, *Tableaux, clusters et graphes*, du *Manuel de l'utilisateur LabVIEW*.

Comment puis-je créer un graphe à barres ?

L'élément de menu **Tracés communs** de la légende offre le tracé à barres comme un des styles disponibles. Les autres attributs du tracé permettent d'utiliser une ligne de base différente et d'effectuer des tracés à barres horizontales. De plus, consultez `exemples\general\graphs\bargraph.llb`. Cette bibliothèque de VIs contient un exemple de graphe à barres et deux VIs pour créer des graphes à barres à partir d'un tableau d'entrée. Le G Picture Control Toolkit peut également créer des graphes à barres.

Comment puis-je créer des tracés en coordonnées polaires et des graphes déroulants de Smith ?

Le G Picture Control Toolkit contient des exemples de programmes pour créer des tracés en coordonnées polaires et des graphes déroulants de Smith. Le toolkit est un progiciel graphique polyvalent qui permet de créer des faces-avant d'apparence arbitraire. La bibliothèque Picture VI, qui est livrée avec le toolkit, met à disposition un ensemble de commandes courantes en deux dimensions pour construire des images graphiques.

Que dois-je faire pour faire pivoter de 90 degrés une étiquette pour l'axe des Y ?

BridgeVIEW et LabVIEW ne permettent pas de faire pivoter du texte. Pour imposer une rotation à du texte, créez-le d'abord dans une application qui offre cette possibilité, par exemple Paintbrush de Windows, et enregistrez-le dans un format convenable (consultez la section *Personnalisation des commandes avec des graphiques importés* du chapitre 8, *Introduction aux objets de la face-avant*, pour plus d'informations sur l'importation des graphiques). Vous pouvez ensuite importer ce graphique et le placer près de l'axe des Y du graphe ou du graphe déroulant.

Que dois-je faire pour placer une étiquette de texte sur un curseur du graphe ?

Pour un accès par l'intermédiaire d'un programme, vous pouvez utiliser d'abord l'attribut "Cursor Name Visible" pour afficher le nom du curseur. Vous pouvez utiliser ensuite l'attribut "Cursor Name" pour assigner un nom au curseur. Lors d'un usage interactif, le menu de style du curseur permet de contrôler son nom dans la zone du graphe correspondant à l'affichage du curseur.

Une fois que j'ai ajouté un curseur à mon graphe, comment puis-je l'enlever ?

Vous devez vider le tableau qui contient les clusters détenant les informations relatives à chaque curseur. Sélectionnez **Opérations sur les données»Vider le tableau** dans le menu local de la zone d'affichage du curseur, ou envoyez un tableau vide, au moyen d'un programme, vers l'attribut node "Cursor List".

Messages d'erreur et blocages

Que dois-je faire lorsqu'apparaît une boîte de dialogue indiquant que mon application est à court de mémoire ?

LabVIEW et BridgeVIEW allouent la mémoire en fonction des besoins, mais les tableaux et chaînes de caractères doivent être enregistrés dans un bloc de mémoire contigu. Si votre application LabVIEW ou BridgeVIEW ne parvient pas à trouver un bloc de mémoire inutilisée (physique ou virtuelle) suffisamment grand pour la chaîne de caractères ou le tableau, une boîte de dialogue apparaît pour indiquer qu'elle n'a pas été capable d'allouer la mémoire requise.

Si vous voulez enregistrer les modifications apportées à vos VIs mais que l'application est à court de mémoire, vous pouvez enregistrer vos VIs dans un autre endroit ou vous assurer que vous avez une copie de sauvegarde. Une fois que vous redémarrez le logiciel avec davantage de mémoire, vous pouvez charger ces VIs en mémoire, vérifier que l'enregistrement a réussi et continuer le développement de vos VIs.

(Windows) Que dois-je faire si LabVIEW ou BridgeVIEW se bloque pendant l'impression ?

Ce blocage peut être imputable au driver vidéo ou au driver de l'imprimante. Reportez-vous à la question concernant les blocages aléatoires sous Windows, ci-après, pour plus d'informations sur la conduite à suivre.

Pendant l'exécution de LabVIEW ou de BridgeVIEW, que dois-je faire si je reçois un message d'erreur qui comporte un nom de fichier et un numéro de ligne de code source ?

Ceci indique qu'une erreur a eu lieu et que l'application ne peut pas continuer. Veuillez aviser National Instruments de ce message et expliquer l'action qui a déclenché ce message.

(Windows) Que dois-je faire si LabVIEW ou BridgeVIEW a tendance à se bloquer aléatoirement ?

Les blocages peuvent être des défaillances de protection générales ou des messages d'erreur qui comportent un nom de fichier et un numéro de ligne de code source. Souvent, les blocages aléatoires sont dus à des problèmes au niveau du driver vidéo de l'ordinateur. Pour essayer de déterminer si tel est le cas, utilisez le driver VGA standard en tant que driver vidéo. Pour

changer de driver vidéo, sélectionnez VGA dans la liste de drivers vidéo du programme de configuration de Windows. LabVIEW et BridgeVIEW utilisent l'API standard de Windows pour ses appels graphiques ; cependant, de nombreux drivers vidéo ne sont pas complètement conformes à ce standard. Si les blocages n'ont pas lieu avec le driver VGA standard, il est recommandé que vous vous procuriez une version plus récente de votre driver vidéo. Vous pouvez probablement obtenir la dernière version du logiciel du driver vidéo auprès du fabricant de votre PC et auprès du fabricant de la carte vidéo.

Si le problème n'est pas résolu par une mise à jour du driver vidéo, veuillez contacter National Instruments.

(Windows) Que dois-je faire si je reçois une erreur de parité de mémoire, suivie d'une défaillance de protection générale ?

Les erreurs de parité de mémoire sont dues à des problèmes de mémoire au sein de votre machine. Cette mémoire peut être de la mémoire virtuelle, indiquant un problème sur votre disque dur, ou de la mémoire physique, indiquant une barrette SIMM défectueuse. Pour déceler des problèmes sur votre disque dur, utilisez un progiciel utilitaire standard d'analyse de disque, tel que Norton Utilities. Pour localiser les problèmes au niveau de la mémoire vive physique, effectuez physiquement une rotation des barrettes SIMM dans votre machine, en réinitialisant le PC après chaque rotation. Lorsque la barrette SIMM défectueuse est placée dans le banc de mémoire du bas, la machine ne s'initialise pas du tout. Il est alors recommandé de remplacer la barrette SIMM.

Problèmes de plate-forme et compatibilité

Quelles sont les conditions nécessaires pour transférer des VIs d'une plate-forme à une autre ?

Lorsqu'un fichier de VI est porté vers une autre plate-forme, aucune conversion n'est nécessaire pour que LabVIEW ou BridgeVIEW puisse le lire. Lorsqu'un VI est ouvert, il est recompilé pour la nouvelle plate-forme. Cela signifie que vous *devez* inclure le diagramme d'un VI si vous voulez l'utiliser sur une autre plate-forme. Pour les VIs contenant des CIN, vous devez recompiler les CIN sur la nouvelle plate-forme et refaire ensuite une édition de liens pour les rattacher au VI porté. Éliminez les fonctions spécifiques à une plate-forme (par exemple, Apple Events pour Macintosh, DDE pour Windows, et ainsi de suite) avant de porter un VI vers une autre plate-forme.

Vous pouvez transférer des VIs d'une plate-forme à une autre par l'intermédiaire de réseaux, de modems ou de disquettes. Les VIs sont enregistrés dans le même format de fichier sur toutes les plates-formes. Si vous transférez des VIs sur un réseau à l'aide de ftp ou d'un modem, veuillez à spécifier un transfert binaire.

Si le transfert s'effectue par des disquettes, des utilitaires de conversion de disques sont nécessaires pour lire les disques provenant d'autres plates-formes. Les utilitaires de conversion changent le format des fichiers enregistrés sur le disque en un format différent. La plupart des utilitaires de conversion de fichiers permettent non seulement de *lire* les fichiers provenant d'une autre plate-forme, mais également d'*écrire* des fichiers dans le format de disque de cette plate-forme. Par exemple, il y a des utilitaires tels que MacDisk et TransferPro qui sont disponibles pour PC et qui transfèrent les disquettes Macintosh au format PC et vice versa. Sur Macintosh, DOS Mounter et Apple File Exchange sont deux utilitaires qui convertissent les fichiers résidant sur des disquettes au format DOS au format Macintosh et vice versa. Pour Sun et HP, il y a PC File System (PCFS) qui aide SunOS et HP-UX à lire et à écrire des disquettes au format DOS.

Pour rendre plus aisé le port d'une plate-forme à une autre, vous pouvez enregistrer vos VIs dans une bibliothèque de VIs. Pour faciliter l'enregistrement de tous vos VIs dans une bibliothèque, vous pouvez sélectionner l'option **Distribution pour développement** à partir de la boîte de dialogue **Enregistrer avec options**. De cette manière, vous pouvez enregistrer tous les VIs n'appartenant pas à `vi.lib`, les commandes et les sous-programmes externes dans une bibliothèque unique.

Impression

Que dois-je faire pour imprimer une commande unique à partir de la face-avant (par exemple, un graphe) ?

Pour imprimer seulement un graphe à partir de la face-avant, créez un sous-VI avec un graphe sur sa face-avant.

1. Changez le graphe d'indicateur à commande.
2. Ouvrez le sous-VI et sélectionnez **Exécution»Imprimer à la fin de l'exécution**.
3. Assignez au sous-VI un connecteur et transférez les données du graphe qui se trouve sur le VI principal au graphe qui se trouve sur le sous-VI.

A chaque fois que votre VI principal appelle le sous-VI, il imprime automatiquement le graphe.

Comment puis-je imprimer une chaîne de caractères ?

Utilisez le VI "Initialiser le port série" (`Serial Port Init.vi`) pour initialiser le port auquel est connectée l'imprimante (LPT1, LPT2, et ainsi de suite, sur PC, ou le port de l'imprimante sur Mac), puis le VI "Ecrire sur le port série" (`Serial Port Write.vi`) pour écrire la chaîne de caractères sur le port initialisé. L'imprimante reçoit les données au niveau du port et les imprime. Pour ce faire, vous avez généralement besoin de certaines connaissances du langage de commande de votre imprimante, mais cela fonctionne bien pour de nombreuses applications développées par des utilisateurs de LabVIEW et BridgeVIEW. Si vous êtes un utilisateur de LabVIEW, consultez l'Annexe B, *Questions courantes concernant le G*, du *Manuel de l'utilisateur LabVIEW* pour plus de détails.

(Windows et UNIX) Utilisez le VI "Exec. système" (`System Exec.vi`) pour imprimer un fichier par l'intermédiaire d'une fonction de ligne de commande. Le VI est situé dans **Fonctions»Communication**. Par exemple, sous Windows NT/95, si vous voulez imprimer un document contenant du texte, utilisez le VI "Exec. système" (`System Exec.vi`) avec la commande suivante :

```
notepad /p <nom du document>
```

Le bloc-notes a une limite de 32 Ko. Pour des documents plus volumineux, utilisez un éditeur de texte différent.

(Macintosh) Vous pouvez utiliser le VI “AESend Print Document” pour indiquer à l’autre application d’imprimer un document. Le VI est situé dans **Fonctions»Communication»AppleEvent**. Pour plus de détails, consultez la section *Autres méthodes d’impression* du chapitre 5, *Impression et documentation des VIs*. Vous pouvez également utiliser l’option d’impression contrôlée par un programme, comme cela est décrit dans la question précédente.

Que dois-je faire pour imprimer toutes les données sur un graphe déroulant ?

Le graphe déroulant imprime uniquement les données affichées lorsque vous imprimez la face-avant. Pour imprimer toutes les données qui figurent dans un graphe déroulant, y compris celles qui se trouvent dans le buffer de l’historique, faites d’abord une mise à l’échelle automatique de l’axe des x du graphe déroulant en ouvrant son menu local, en faisant apparaître la palette du graphe déroulant et en verrouillant l’axe des x , ou en ajustant l’axe des x au moyen d’un programme par l’intermédiaire d’un attribut node de graphe déroulant.

Pourquoi LabVIEW ou BridgeVIEW se bloque-t-il lors de l’impression ?

Sous Windows, ce blocage peut être dû au driver vidéo ou au driver de l’imprimante.

Reportez-vous à la question concernant les blocages aléatoires sur votre plate-forme, dans la section *Messages d’erreur et blocages* de cette annexe, pour plus d’informations sur la conduite à tenir.

Comment utiliser l’impression PostScript ?

Si votre imprimante fonctionne avec le langage PostScript, installez et sélectionnez le driver d’imprimante PostScript pour votre imprimante. Dans LabVIEW ou BridgeVIEW, ouvrez la boîte de dialogue **Edition»Préférences»Impression** et sélectionnez **Impression PostScript**.

Si vous avez une imprimante PostScript, vous pouvez profiter des avantages suivants :

- Les impressions PostScript reproduisent plus fidèlement l’image de l’écran.
- PostScript reproduit plus fidèlement les motifs et les styles de lignes.

Lorsque je sélectionne l'impression PostScript, pourquoi est-ce que mon impression se traduit par un entassement de texte imprimé en haut de la page ?

L'impression traduit les données à imprimer du VI au format PostScript (.ps) et les envoie au driver de l'imprimante Windows sous forme de texte PostScript. Si le driver de votre imprimante n'est pas prévu pour une impression PostScript, ce qui est imprimé est en fait le texte même du fichier PostScript, au lieu d'une image graphique. Reportez-vous à la question précédente concernant l'impression PostScript pour plus d'informations.

Que faire si le texte sur les étiquettes et les commandes de la face-avant est coupé lors de l'impression du VI ?

Cela se produit lorsque la taille de la fonte de l'imprimante ne correspond pas à la taille de la fonte sur le moniteur.

- Si votre imprimante fonctionne avec le langage PostScript, utilisez le driver d'imprimante PostScript pour votre imprimante pour permettre des impressions de meilleure qualité. Reportez-vous à la question ci-dessus pour plus d'informations sur l'impression PostScript.
- Agrandissez les étiquettes et les commandes de la face-avant de manière à ce que, lorsque le texte se dilate à cause d'une mauvaise concordance de fontes, le texte tienne toujours à l'intérieur des limites de la commande ou de l'étiquette.
- Trouvez une fonte qui soit de la même taille sur le moniteur et sur l'imprimante. De nombreux drivers d'imprimante ont des algorithmes de substitution de fonte pour faciliter ce processus.
- **(Windows)** Utilisez l'impression bitmap, qui reproduit exactement sur le papier ce que vous voyez sur le moniteur.

(Windows) Comment sélectionner l'impression bitmap ?

Ouvrez la boîte de dialogue **Edition»Préférences** et sélectionnez **Impression**. La dernière option sur l'écran est **Impression bitmap**.

Divers

Qu'est-ce qu'Info-LabVIEW et comment puis-je m'abonner ?

Info-LabVIEW est un réseau consacré à LabVIEW et à ses utilisateurs, subventionné et soutenu par les utilisateurs. Ceux-ci peuvent envoyer des informations à une adresse électronique spécifique (voir ci-dessous) ; ce courrier est ensuite diffusé à tous les utilisateurs abonnés à la liste. D'autres utilisateurs pourront alors adresser une réponse au groupe ou peut-être directement à la personne qui a écrit le message.

Plusieurs utilisateurs chevronnés, ainsi que certains employés de National Instruments, sont abonnés à la liste et répondent à des questions diverses. Globalement, le groupe Info-LabVIEW a été extrêmement bien accueilli. Bien que le groupe ne soit pas connecté à National Instruments, des réponses officielles sont occasionnellement adressées au groupe. La plupart du temps, les conversations ont lieu strictement entre utilisateurs.

Si vous voulez vous abonner à la liste, envoyez un courrier électronique à cette adresse :

`info-labview-request@pica.army.mil`

Il y a deux manières pour les utilisateurs de recevoir les messages : le format standard et le format sommaire. Le format standard signifie que vous recevez les messages à mesure qu'ils sont émis. Le format sommaire signifie que vous recevez un colis à la fin de chaque journée, contenant tous les messages de la journée. Indiquez que vous désirez le format sommaire dans les messages électroniques que vous envoyez à `info-labview-request` si vous voulez recevoir un message global par jour.

Une fois que vous êtes abonné à la liste, vous recevez des messages provenant d'autres abonnés. Le volume des échanges au sein du groupe est relativement soutenu, de 10 à 30 messages par jour. Pour ajouter des informations à la liste, envoyez un courrier électronique à l'adresse suivante :

`info-labview@pica.army.mil`

Comment puis-je charger et exécuter dynamiquement des VIs ?

Lorsqu'un sous-VI se trouve dans le diagramme d'un autre VI, le sous-VI se charge en mémoire dès que le VI appelant se charge en mémoire. Pour répondre à des considérations de mémoire, il peut vous être utile de charger et de décharger dynamiquement des VIs en mémoire lors de l'exécution de votre programme. Pour ce faire, vous pouvez utiliser les fonctions VI Serveur. Pour plus d'informations, consultez le chapitre 21, *VI Serveur* de ce manuel.

(Macintosh) En plus du VI Serveur, qui est indépendant de la plate-forme, vous pouvez utiliser les Apple Events sur Macintosh pour charger et exécuter dynamiquement des VIs. Les VIs sont situés dans la palette **Fonctions»Communication»AppleEvent** ou dans sa sous-palette **Apple Events spécifiques à LabVIEW**, et ils s'appellent :

- AESend Finder Open
- AESend Open, Run, Close VI
- AESend Run VI
- AESend Close VI

Comment puis-je remplacer un sous-VI par un autre VI portant le même nom ?

LabVIEW et BridgeVIEW désignent tous les VIs par leur nom et, par conséquent, ils ne chargent jamais deux VIs du même nom, même si les chemins menant à ces VIs sont différents. Ceci est valable également pour les sous-VIs. Lorsque vous sélectionnez **Remplacer** sur un sous-VI pour le remplacer par un VI du même nom, le G réalise qu'il a déjà une copie du VI en mémoire et il remplace le VI par lui-même.

Pour charger la nouvelle copie du sous-VI, vous devez d'abord enlever l'ancienne copie de la mémoire. Les VIs courants figurent sous le menu **Fenêtres**. La manière la plus simple d'enlever le sous-VI de la mémoire est de fermer le sous-VI et tous les autres VIs qui appellent le sous-VI. Ouvrez la nouvelle copie du sous-VI pour la charger en mémoire, vérifiez **Fichier»Infos sur le VI...** pour confirmer qu'il s'agit bien de la version que vous voulez, puis rouvrez le VI principal. Lorsque le VI essaie d'appeler le sous-VI, il trouve une copie en mémoire (la nouvelle version) et il l'utilise plutôt que de chercher la version précédente.

La solution à beaucoup de ces problèmes est de toujours donner aux sous-VIs des noms uniques.

Le chargement de deux VIs qui appellent deux sous-VIs distincts portant le même nom pose des problèmes.

Par exemple, le VI “Principal1” (`main1.vi`) et le VI “Principal2” (`main2.vi`) appellent tous deux des sous-VIs distincts qui sont tous deux appelés VI “Sous-VI” (`subVI.vi`).

1. Le VI “Principal1” (`main1.vi`) se charge en mémoire, et le VI “Sous-VI” (`subVI.vi`) se charge parce qu’il est appelé par le VI “Principal1” (`main1.vi`).
2. Le VI “Principal2” (`main2.vi`) se charge en mémoire, et les informations de l’éditeur de liens indiquent à LabVIEW ou BridgeVIEW de charger le VI “Sous-VI” (`subVI.vi`). Si le cadre connecteur est exactement le même, il effectue probablement le lien (mais le comportement pourra être très inattendu). Si le cadre connecteur est différent, vous recevez une erreur indiquant que la liaison est mauvaise.

Solution : donnez toujours aux sous-VIs des noms uniques.

Comment l’exécution des VIs est-elle affectée par la priorité ?

Le système d’exécution du G peut exécuter plusieurs VIs ou sous-diagrammes en parallèle. Cinq priorités peuvent être assignées aux VIs : arrière-plan (la plus faible), normale, supérieure à la normale, élevée et maximale (la plus élevée). Une autre priorité, la priorité de type sous-programme, force un VI à s’exécuter jusqu’à la fin sans partager son thread du système d’exécution avec d’autres VIs. Chaque système d’exécution a une file d’attente pour les sections de code des VIs qui se trouvent dans le système. La file d’attente est classée par ordre de priorité, de sorte que les VIs à priorité plus élevée s’exécutent avant les VIs à priorité plus faible. Tant que des éléments à priorité élevée sont dans la file d’attente, les éléments à priorité plus faible ne peuvent pas atteindre le début de la file d’attente et ne peuvent pas s’exécuter. Cela signifie qu’il est possible que des VIs à faible priorité ne parviennent pas à s’exécuter si des VIs à priorité plus élevée s’exécutent de façon continue sans interruption pour un timeout ou une autre activité asynchrone. Pour cette raison, prenez garde lorsque vous augmentez la priorité d’un VI.

La file d’attente d’exécution du G a des points d’entrées pour chacune des cinq priorités. Il n’y a pas de point d’entrée pour la priorité de type sous-programme. Un VI à priorité de type sous-programme commence à s’exécuter lorsqu’il est appelé par un VI, et il domine son thread de système d’exécution jusqu’au retour à un VI appelant affecté d’une autre priorité. Il

peut appeler d'autres VIs à priorité de type sous-programme, mais pas des VIs affectés d'une autre priorité. Les VIs à priorité de type sous-programme ne mettent pas à jour leurs indicateurs, ne peuvent pas être mis en attente ni effectuer d'autres opérations asynchrones et, par conséquent, ils ne peuvent jamais être placés dans une file d'attente. De ce point de vue, une fois qu'ils commencent à s'exécuter, ce sont les VIs à priorité la plus élevée dans un système d'exécution.

Dans les systèmes multithread, il y a plusieurs systèmes d'exécution, chacun comportant une file d'attente d'exécution et un ou plusieurs threads. Les threads sont affectés d'une priorité d'exécution, si bien que le système d'exploitation peut placer un thread à faible priorité en attente lorsqu'un thread à priorité plus élevée est prêt pour l'exécution. La priorité des VIs est utilisée pour déterminer la file d'attente du système d'exécution dans laquelle il est préférable de les placer. Un VI à priorité de type sous-programme qui est appelé dans un système multithread domine le thread dans lequel il est appelé mais ne peut pas empêcher d'autres VIs dans d'autres threads de s'exécuter. Cela signifie que vous ne pouvez pas utiliser des VIs à priorité de type sous-programme pour bloquer l'exécution de tout autre VI sur des systèmes comportant d'autres threads à priorité supérieure ou égale.

Sur les systèmes multithread, l'interface utilisateur est un thread à priorité normale. Sur certains systèmes, elle peut être reléguée derrière des VIs qui s'exécutent à une priorité plus élevée et qui ne s'arrêtent jamais pour un timeout ou pour effectuer une autre activité asynchrone. Il peut être impossible de changer une commande pour arrêter un VI à priorité élevée qui ne permet jamais à l'interface utilisateur de prendre la commande en compte. Pour cette raison, prenez garde lorsque vous augmentez la priorité d'un VI.

Comment des interruptions peuvent-elles être gérées par le G sans utiliser d'interrogation ?

Les occurrences constituent une méthode dont dispose le G pour imposer à une section de code d'attendre un événement particulier. Par exemple, vous pouvez écrire un CIN qui lance un processus (sous Windows, par l'intermédiaire d'une DLL, par exemple) qui génère une occurrence lorsqu'un événement a lieu dans la DLL. Une section particulière du code du diagramme s'exécute, et une autre section ou un VI attend l'événement. Lorsque l'événement survient, il enregistre l'occurrence dans LabVIEW ou BridgeVIEW, et le code approprié commence à s'exécuter dès que vient son tour dans la file d'attente.

En interne, LabVIEW et BridgeVIEW examinent la file d'attente lorsqu'elle termine un bloc de code atomique. Cela signifie que, si une autre exécution est en cours, cette exécution doit s'achever avant qu'il soit possible de traiter l'occurrence, et donc l'interruption.



Remarque

Il n'y a pas vraiment de limite concernant le temps que cela peut prendre, en particulier si le VI contient des CIN, qui peuvent mettre très longtemps à terminer leur exécution.

Avec des boucles While imbriquées, comment puis-je arrêter les deux boucles sans que rien dans la boucle extérieure ne s'exécute après l'arrêt de la boucle intérieure ?

Mettez le code que vous ne voulez pas exécuter lors de la dernière itération dans une structure Condition. La meilleure solution est que le terminal de condition de la boucle intérieure soit connecté au terminal de sélection de la structure Condition.

Qu'est-ce que l'ordonnement de la face-avant ?

Edition»Ordonnement de la face-avant détermine l'ordonnement des objets sur la face-avant. En d'autres termes, si vous utilisez la touche <Tab> du clavier pour passer d'un objet à un autre, le focus de la touche passe d'un objet à un autre en suivant l'ordre déterminé par l'ordonnement de la face-avant. Seules des commandes peuvent recevoir le focus de la touche ; la touche <Tab> ne permet pas de faire passer le focus sur les indicateurs. Par défaut, l'ordonnement de la face-avant est l'ordre dans lequel vous créez les objets sur la face-avant.

En ce qui concerne l'enregistrement des données de la face-avant, l'ordonnement de la face-avant détermine l'ordre dans lequel les différents objets sont compactés dans un cluster au sein du fichier.

Comment puis-je indiquer à LabVIEW ou à BridgeVIEW de lancer automatiquement un VI particulier ?

Par défaut, LabVIEW et BridgeVIEW lancent et créent un VI "Sans-titre1" (Untitled1.vi). Vous pouvez souhaiter que votre application lance automatiquement un VI particulier. Vous pouvez aussi configurer le VI pour qu'il s'exécute lorsqu'il est chargé, de sorte que le lancement de LabVIEW ou BridgeVIEW non seulement ouvre votre VI, mais aussi commence à l'exécuter. Sélectionnez les **Options d'exécution** appropriées dans **Configuration du VI...** pour que le VI s'exécute quand il est ouvert. Si vous spécifiez une bibliothèque (.lib) à ouvrir lors du lancement de votre application, cela ouvre tous les VIs du **Niveau principal**. Utilisez

Fichier»Editer une bibliothèque de VIs... pour affecter des VIs au **Niveau principal**. Enfin, si vous spécifiez une bibliothèque et qu'aucun VI ne porte la mention **Niveau principal**, au moment du lancement, une boîte de dialogue de fichier invite l'utilisateur à sélectionner un VI dans la bibliothèque spécifiée.

Pour indiquer à LabVIEW ou à BridgeVIEW de lancer un VI particulier, suivez les instructions ci-dessous pour la plate-forme qui vous intéresse.

(Windows) Sélectionnez l'icône LabVIEW ou BridgeVIEW, sélectionnez **Fichier»Propriétés** (dans le Gestionnaire de programmes) et changez le nom de chemin dans **Ligne de commande** de manière à désigner votre VI. Par exemple, pour que LabVIEW charge automatiquement le VI "Test" (test.vi), écrivez la **Ligne de commande** comme suit :

```
c:\labview\labview.exe test.vi
```

Si le VI "Test" (test.vi) se trouve à l'intérieur d'une bibliothèque appelée test.llb, alors :

```
c:\labview\labview.exe "test.llb\test.vi"
```

Il peut être nécessaire de spécifier en entier le chemin menant au VI.

Si vous voulez lancer un VI à partir de la ligne de commande

(Démarrer»Exécuter...), la syntaxe est la suivante :

```
c:\labview\labview.exe <chemin conduisant au VI à partir  
du répertoire labview>
```

Par exemple, pour lancer le VI "Lisez-moi" (readme.vi) se trouvant dans le répertoire c:\labview\examples, la commande est la suivante :

```
c:\labview\labview.exe examples\readme.vi
```

Si le VI se trouve sur un chemin différent, vous devez spécifier en entier le chemin menant au VI :

```
c:\labview\labview.exe c:\coolapp\mycool.vi
```

Si un répertoire dans le chemin contient des espaces, vous devez mettre le chemin entre guillemets :

```
c:\labview\labview.exe "c:\cool application\mycool.vi"
```

Si le VI se trouve dans une LLB, vous pouvez au choix :

1. Spécifier le chemin entier entre guillemets :

```
c:\labview\labview.exe "c:\coolapp\eagle.llb\mycool.vi"
```

ou

2. Ouvrir la Configuration du VI et sélectionner **Exécuter lors de l'ouverture**. Ensuite, sélectionnez **Fichier»Editer une bibliothèque de VIs...** et affectez le VI au **Niveau principal**.

Dans ce cas, vous avez seulement besoin de spécifier le chemin menant à la llb :

```
c:\labview\labview.exe "c:\coolapp\eagle.llb"
```

Si vous voulez que LabVIEW lance automatiquement un VI, modifiez les propriétés de Raccourci et spécifiez le chemin dans Destination, comme cela est illustré ci-dessus.

(Macintosh) Il n'y a aucun moyen d'indiquer à l'application LabVIEW de lancer automatiquement un VI particulier sur Macintosh ; cependant, vous pouvez lancer un VI en double-cliquant sur son icône dans le Finder. Si vous avez plusieurs copies de LabVIEW installées sur votre machine, le Finder détermine la copie qui est lancée lorsque vous double-cliquez sur le VI (vous ne pouvez pas contrôler quelle copie est lancée). Par exemple, le Finder peut lancer la version exécutable si la version exécutable et la version de développement complète sont toutes deux installées. Si vous utilisez le Système 7 ou une version ultérieure, vous pouvez utiliser le glisser-déposer pour lancer un ou plusieurs VIs ou bibliothèques de VIs.

(UNIX) LabVIEW répond aux options de la ligne de commande pour lancer un VI particulier lorsqu'il est ouvert. Par conséquent, vous pouvez taper :

```
labview /usr/home/test.vi
```

ou

```
labview /usr/home/test.llb/test.vi
```

pour lancer le VI "Test" (test.vi), suivant le chemin correct. Vous pouvez utiliser un script simple pour créer une commande qui lance LabVIEW avec un VI particulier.

Comment puis-je changer au moyen d'un programme les entrées d'un type énuméré ?

Vous ne pouvez pas changer le type (les chaînes de caractères) d'un type de données énuméré au moyen d'un programme, tout comme vous ne pouvez pas changer au moyen d'un programme une commande d'entier en commande de double précision ou une commande de chaîne de caractères en commande de chemin. Les chaînes de caractères d'un Enum font partie de son type de données et, par conséquent, elles peuvent seulement être modifiées lors d'une édition. Il est possible de lire les chaînes de caractères de l'Enum grâce à un attribute node, mais vous ne pouvez pas les écrire par l'intermédiaire d'un attribute node.

Si vous voulez changer au moyen d'un programme les valeurs de texte de l'Enum, utilisez plutôt une commande de menu déroulant. Vous pouvez utiliser la commande de menu déroulant pour lire et écrire des chaînes de caractères au moyen d'un programme par l'intermédiaire de l'attribut Strings[].

Comment puis-je masquer les barres de menus d'un VI ?

Tous les attributs des VIs, y compris l'affichage de la barre de menus et de la barre d'outils, sont déterminés par l'intermédiaire de la boîte de dialogue dans **Configuration du VI...** Pour accéder à la boîte de dialogue de Configuration du VI, ouvrez le menu local de l'icône du VI dans le coin supérieur droit de la face-avant et sélectionnez **Configuration du VI...** Consultez le chapitre 6, *Configuration des VIs et sous-VIs*, pour plus de détails.

Comment puis-je désactiver les interruptions de la souris pour améliorer les performances ?

Les interruptions causées par le déplacement de la souris ou lorsque l'on clique avec la souris sur un élément coûtent du temps au processeur. Dans certaines applications pour lesquelles le temps est un paramètre classique, cette activité d'interruption peut résulter en une perte de données ou donner lieu à d'autres problèmes avec l'application. Outre débrancher la souris, il n'y a aucune méthode recommandée pour désactiver les interruptions dues à la souris. En veillant à ce que la souris ne bouge pas pendant les portions de votre application pour lesquelles le temps est essentiel, vous pouvez minimiser ses effets.



Informations à l'attention du client

Cette annexe contient les formulaires qui vous permettront de réunir les renseignements nécessaires pour vous aider à résoudre vos problèmes techniques, ainsi qu'une fiche que vous pouvez utiliser pour nous signaler vos remarques sur nos documentations de produits. Lorsque vous nous contactez, veuillez nous fournir les renseignements contenus dans le formulaire de support technique et, le cas échéant, dans la fiche de configuration de votre système de façon à ce que nous puissions répondre à vos questions dans les plus brefs délais.

National Instruments offre une assistance technique par téléphone, télécopie et par l'intermédiaire de systèmes électroniques pour que vous puissiez obtenir rapidement l'information que vous souhaitez. Nos services de support électroniques comprennent un BBS, un site FTP, un système Fax-on-demand et un service de courrier électronique. Si vous rencontrez des problèmes de matériel ou de logiciel, essayez tout d'abord les services de support électroniques. Si les renseignements fournis par ces systèmes sont insuffisants, nos centres de support technique pourvus d'ingénieurs d'application, offrent également un support téléphonique et par télécopie.

Services électroniques

Support par site FTP

Pour accéder à notre site FTP, ouvrez une session sur l'hôte Internet, `ftp.natinst.com`, comme anonyme et utilisez votre adresse Internet, par exemple `pauldupont@ailleurs.com`, comme mot de passe. Les fichiers et documents de support se trouvent dans les répertoires `/support`.

Support par le système Fax-on-Demand

Le système Fax-on-Demand est un système de recherche documentaire accessible 24 heures sur 24 contenant une bibliothèque documentaire portant sur une vaste gamme d'informations techniques. Vous pouvez accéder au système Fax-on-Demand avec un téléphone à touches en composant le 512 418 1111.

Support par courrier électronique (disponible aux Etats-Unis seulement)

Vous pouvez poser vos questions techniques à l'équipe d'ingénieurs d'applications par courrier électronique à l'adresse Internet indiquée ci-dessous. N'oubliez pas de faire figurer vos nom, adresse et numéro de téléphone, de façon à ce que nous puissions vous communiquer des solutions et suggestions.

`support@natinst.com`

Support par affichage électronique

National Instruments dispose de sites BBS et FTP spécialisés en support technique accessible 24 heures sur 24, contenant un large éventail de fichiers et de documents pour répondre aux questions les plus courantes de ses clients. Ces sites vous permettent également de télécharger les tout derniers drivers d'instruments, mises à jour et programmes d'exemple. Pour obtenir des instructions enregistrées sur la façon d'utiliser les services BBS et FTP, et pour obtenir les services d'informations automatisés BBS, composez le 512 795 6990. Vous pouvez accéder à ces services de la manière suivante :

En France : 01 48 65 15 59

Jusqu'à 9,600 bauds, 8 bits de données, 1 bit d'arrêt, aucune parité

Aux Etats-Unis : 512 794 5422

Jusqu'à 14,400 bauds, 8 bits de données, 1 bit d'arrêt, aucune parité

Au Royaume-Uni : 01635 551422

Jusqu'à 9,600 bauds, 8 bits de données, 1 bit d'arrêt, aucune parité

Support par téléphone et télécopie

National Instruments dispose de bureaux répartis dans le monde entier. Utilisez la liste suivante pour localiser le numéro de support technique de votre pays. Si National Instruments n'a pas de bureau de représentation dans votre pays, contactez le revendeur de votre logiciel pour obtenir de l'aide.

Pays	Téléphone	Télécopie
Allemagne	089 741 31 30	089 714 60 35
Australie	03 9879 5166	03 9879 6277
Autriche	0662 45 79 90 0	0662 45 79 90 19
Belgique	02 757 00 20	02 757 03 11
Bésil	011 288 3336	011 288 8528
Canada (Ontario)	905 785 0085	905 785 0086
Canada (Québec)	514 694 8521	514 694 4399
Corée	02 596 7456	02 596 7455
Danemark	45 76 26 00	45 76 26 02
Espagne	91 640 0085	91 640 0533
Etats-Unis	512 795 8248	512 794 5678
Finlande	09 725 725 11	09 725 725 55
France	01 48 14 24 24	01 48 14 24 14
Hong Kong	2645 3186	2686 8505
Israël	03 6120092	03 6120095
Italie	02 413091	02 41309215
Japon	03 5472 2970	03 5472 2977
Mexique	5 520 2635	5 520 3282
Norvège	32 84 84 00	32 84 86 00
Pays-bas	0348 433466	0348 430673
Royaume-Uni	01635 523545	01635 523154
Singapour	2265886	2265887
Suède	08 730 49 70	08 730 43 70
Suisse	056 200 51 51	056 200 51 55
Taiwan	02 377 1200	02 737 4644

Formulaire de support technique

Photocopiez ce formulaire et mettez-le à jour chaque fois que vous faites des changements d'ordre logiciel ou matériel, et utilisez la copie de ce formulaire complété comme référence pour votre configuration actuelle. Complétez ce formulaire avec soin avant de contacter National Instruments pour obtenir le support technique nécessaire, afin de permettre à nos ingénieurs d'application de répondre plus efficacement à vos questions.

Si vous utilisez des produits logiciels et matériels de National Instruments ayant rapport au problème, ajoutez les fiches de configuration se trouvant dans les manuels de l'utilisateur correspondants. Ajoutez toute page supplémentaire si nécessaire.

Nom _____

Société _____

Adresse _____

Téléphone (___) _____ Télécopie (___) _____

Marque d'ordinateur _____ Modèle _____ Processeur _____

Système d'exploitation (inclure le numéro de version) _____

Fréquence d'horloge _____ MHz RAM _____ Mo Carte graphique _____

Souris ___oui ___non Autres cartes installées _____

Capacité du disque dur _____ Mo Marque _____

Instruments utilisés _____

Modèle de produit matériel de National Instruments _____ Révision _____

Configuration _____

Produit logiciel de National Instruments _____ Version _____

Configuration _____

Description du problème : _____

Listez tout message d'erreur : _____

Le problème apparaît chaque fois que les opérations suivantes sont effectuées : _____

Fiche de configuration logicielle et matérielle LabVIEW

Inscrivez les paramètres et les modifications apportées à votre matériel et à votre logiciel. Complétez une nouvelle copie de ce formulaire chaque fois que vous modifiez la configuration de votre logiciel ou de votre matériel, et utilisez ce formulaire comme référence pour votre configuration actuelle. Complétez ce formulaire avec soin avant de contacter National Instruments pour obtenir un support technique afin de permettre à nos ingénieurs d'application de répondre plus efficacement à vos questions.

Produits de National Instruments

Matériel DAQ _____

Niveau d'interruption du matériel (IRQ) _____

Canaux DMA du matériel _____

Adresse de base d'E/S du matériel _____

Choix de programmation _____

Version NI-DAQ ou LabVIEW _____

Autres cartes dans le système _____

Adresse de base d'E/S des autres cartes _____

Canaux DMA des autres cartes _____

Niveau d'interruption des autres cartes _____

Autres produits

Marque et modèle de l'ordinateur _____

Microprocesseur _____

Fréquence ou vitesse d'horloge _____

Type de carte vidéo installée _____

Version du système d'exploitation _____

Mode du système d'exploitation _____

Langage de programmation _____

Version du langage de programmation _____

Autres cartes présentes dans le système _____

Adresse de base d'E/S des autres cartes _____

Canaux DMA des autres cartes _____

Niveau d'interruption des autres cartes _____

Vos commentaires sur la documentation...

National Instruments vous encourage à nous signaler vos remarques sur les documentations livrées avec nos produits. Ces informations nous aideront à vous garantir la qualité que vous attendez de nos produits.

Titre : *Manuel de référence de programmation en G*

Date d'édition : Juillet 1998

Référence : 321982A-01

Veillez apporter des commentaires sur l'exhaustivité, la clarté et l'organisation de ce manuel :

Si vous avez trouvé des erreurs, indiquez les numéros de pages correspondants et décrivez la nature de ces erreurs :

Nous vous remercions de votre contribution.

Nom _____

Fonction _____

Société _____

Adresse _____

Adresse électronique _____

Téléphone (____) _____ Télécopie (____) _____

A envoyer à : National Instruments
Centre d'Affaires Paris-Nord
Immeuble "Le Continental"
BP 217
93153 Le Blanc-Mesnil Cedex

A télécopier au : numéro de fax de
National Instruments France
01 48 14 24 14

Glossaire

Préfixe	Signification	Valeur
m-	milli-	10^{-3}
μ -	micro-	10^{-6}
n-	nano-	10^{-9}

Symboles

∞	Infini.
Δ	Delta. Différence. Δx désigne la valeur dont x change d'un indice au suivant.
π	Pi.

A

à une dimension	Possédant une dimension, comme dans le cas d'un tableau qui a seulement une ligne d'éléments.
à deux dimensions	Possédant deux dimensions, comme un tableau ayant des lignes et des colonnes.
acquisition de données	DAQ. Processus d'acquisition de données, généralement par l'intermédiaire de cartes enfichables d'entrée numérique ou analogique.
adapter	Changer le descripteur de type d'un élément de données sans affecter l'image mémoire des données.
ANSI	American National Standards Institute (Institut national des normes, aux Etats-Unis).
ASCII	American Standard Code for Information Interchange (code américain standard pour l'échange d'informations).
ATE	Automated Test Equipment (Equipement de test automatisé).

Attribute Node	Nœud de diagramme spécial que vous pouvez utiliser pour contrôler l'apparence et la fonctionnalité des commandes et indicateurs.
auto-indexation	Capacité des structures de boucle à désassembler et assembler des tableaux à leurs bords. Lorsqu'un tableau entre dans une boucle alors que l'auto-indexation est active, la boucle le désassemble automatiquement, c'est-à-dire que les scalaires sont extraits des tableaux à une dimension, les tableaux à une dimension sont extraits des tableaux à deux dimensions, et ainsi de suite. A la fin de leur exécution, les boucles rassemblent les données dans des tableaux en suivant la procédure inverse.

B

barre de menus	Barre horizontale qui contient les noms des menus principaux.
barre d'outils	Barre contenant des boutons de commande que vous pouvez utiliser pour exécuter et mettre au point des VIs.
bibliothèque de VIs	Fichier spécial qui contient une collection de VIs apparentés pour un usage spécifique.
BNF	Backus-Naur Form. Représentation courante pour la grammaire des langages en informatique.
boîte de calcul	Nœud qui exécute des formules que vous entrez sous forme de texte. Particulièrement utile pour les formules longues et trop complexes à représenter sous forme de diagramme.
boîte de description	Documentation en ligne pour un objet du G.
boîte de dialogue	Ecran interactif contenant des messages, dans lequel vous pouvez donner des informations supplémentaires nécessaires pour exécuter une commande.
boucle For	Structure de boucle itérative qui exécute son sous-diagramme un nombre de fois défini. Equivalent au code conventionnel : <code>For i=0 to n-1, do</code>
boucle While	Structure de boucle qui répète une section de code jusqu'à ce qu'une condition soit remplie. Comparable à une boucle Do ou à une boucle Repeat-Until dans les langages de programmation conventionnels.

C

cadre connecteur	Zone située dans l'angle supérieur droit de la fenêtre d'une face-avant et qui affiche les terminaux du VI. Il est sous-jacent au cadre icône.
cadre icône	Zone dans l'angle supérieur droit de la face-avant et du diagramme qui affiche l'icône du VI.
capteur	Périphérique qui produit une sortie de tension ou de courant représentant la mesure d'une propriété physique, telle que la vitesse, la température ou le flux.
caractères non affichables	Caractères ASCII qui ne peuvent pas être affichés, tels que les retours à la ligne, les tabulations, etc.
chemin absolu	Chemin de fichier ou répertoire qui décrit sa position par rapport au niveau principal du système de fichiers.
CIN	<i>Voir</i> Code Interface Node.
classe d'application de VI	Une référence à une application capable de charger des instruments virtuels qui donnent accès aux propriétés et méthodes de l'application.
classe de VI	Une référence à un instrument virtuel qui donne accès aux propriétés et méthodes du VI.
clonage	Faire une copie d'une commande ou d'un autre objet G en cliquant sur le bouton de la souris tout en appuyant sur la touche <Ctrl> (Windows), <option> (Macintosh), <meta> (Sun) ou <Alt> (HP-UNIX) et en faisant glisser la copie jusqu'à son nouvel emplacement. (UNIX) Vous pouvez aussi cloner un objet en cliquant dessus avec le bouton du milieu de la souris et en faisant glisser la copie jusqu'à son nouvel emplacement.
cluster	Ensemble ordonné et non indexé d'éléments de données de n'importe quel type, y compris des nombres, des booléens, des chaînes de caractères, des tableaux ou des clusters. Les éléments doivent être tous des commandes ou tous des indicateurs.
code interface node (CIN)	CIN. Nœud de diagramme spécial par l'intermédiaire duquel vous pouvez lier du code textuel conventionnel à un VI.

coercition	Conversion automatique que le G effectue pour changer la représentation numérique d'un élément de données.
commande	Objet de la face-avant utilisé pour entrer des données dans un VI de façon interactive ou dans un sous-VI par l'intermédiaire d'un programme.
commandes et indicateurs booléens	Objets de la face-avant utilisés pour manipuler et afficher des données booléennes (TRUE ou FALSE). Plusieurs styles sont disponibles : des interrupteurs, des boutons et des LED.
commandes et indicateurs de chaîne de caractères	Objets de la face-avant utilisés pour manipuler et afficher, ou entrer et sortir du texte.
commandes et indicateurs numériques	Objets de face-avant utilisés pour manipuler et afficher, ou entrer et sortir, des données numériques.
commandes PICT personnalisées	Commandes et indicateurs dont des parties peuvent être remplacées par des graphiques et des indicateurs que vous fournissez.
commande de menu déroulant	Commande numérique spéciale qui associe des entiers 32 bits, en commençant par 0 et en augmentant progressivement, avec une série d'étiquettes de texte ou de graphiques.
commande de type graphe	Objet de la face-avant qui affiche des données dans un plan cartésien.
compiler	Processus qui convertit du code de haut niveau en code machine exécutable. Les VIs sont compilés automatiquement avant d'être exécutés pour la première fois après leur création ou après toute modification.
condition	Sous-diagramme d'une structure Condition.
condition de compétition	Se produit lorsque deux ou plusieurs éléments de code qui s'exécutent en parallèle changent la valeur de la même ressource partagée, typiquement une variable globale ou locale.
connecteur	Partie du nœud d'un VI ou d'une fonction qui contient ses terminaux d'entrée et de sortie. Les données sont transmises vers le nœud et reçues de celui-ci par l'intermédiaire d'un connecteur.
constante	<i>Voir</i> constante universelle et constante définie par l'utilisateur.
constante définie par l'utilisateur	Objet de diagramme qui émet une valeur que vous définissez.

constante universelle	Objet de diagramme non éditable qui émet un caractère ASCII particulier ou une constante numérique standard, par exemple, π .
conversion	Changement de type d'un élément de données.
CPU	Central Processing Unit (unité de calcul).

D

dépendance de données	Dans un langage de programmation par flux de données, condition dans laquelle un nœud ne peut pas s'exécuter tant qu'il ne reçoit pas des données d'un autre nœud. <i>Voir aussi</i> dépendance de données artificielle.
dépendance de données artificielle	Condition dans un langage de programmation à flux de données dans laquelle l'arrivée d'une donnée, plutôt que sa valeur, déclenche l'exécution d'un nœud.
descripteur de type de données	Code qui identifie les types de données, utilisé pour le stockage et la représentation des données.
diagramme	Description ou représentation graphique d'un programme ou d'un algorithme. En G, le diagramme, qui est constitué d'icônes exécutables appelées nœuds et de fils de liaison qui transmettent les données entre les nœuds, est le code source du VI.
dimension	Attribut de taille et de structure d'un tableau.
dimensionnement automatique	Redimensionnement automatique des étiquettes pour s'adapter au texte que vous entrez.
données aplaties	Données de n'importe quel type qui ont été converties en une chaîne de caractères, généralement pour être écrites dans un fichier.
driver d'instrument	VI qui commande un instrument programmable.
DUT	Device under test (Périphérique en cours de test).

E

échelle	Partie des commandes et indicateurs à action mécanique, des graphes et des graphes déroulants, qui contient une série de repères ou de points à des intervalles connus pour désigner des unités de mesure.
Editeur d'icône	Interface similaire à celle d'un programme de peinture pour créer des icônes de VIs.
EOF	End-of-File. Offset du caractère de fin de fichier par rapport au début du fichier (c'est-à-dire, l'EOF représente la taille du fichier).
emplacement	Partie inamovible des commandes et indicateurs de face-avant, qui contient des glissières et des échelles.
enregistrement de données	Acquisition et stockage simultanés de données dans un fichier sur disque. Les fonctions d'E/S sur fichier du G peuvent enregistrer des données.
E/S	Entrée/sortie. Transfert de données vers ou à partir d'un système informatique impliquant des canaux de communication, des périphériques d'entrée utilisateur et/ou des interfaces d'acquisition de données et de commande.
étape	Sous-diagramme d'une structure Séquence.
étiquette	Objet de texte utilisé pour nommer ou décrire d'autres objets ou zones sur la face-avant ou le diagramme.
étiquette de nom	L'étiquette d'un objet de face-avant, utilisée pour nommer l'objet et pour le distinguer d'autres objets. Utilisée sur le terminal, les variables locales et les attribut nodes qui font partie de l'objet. <i>Voir aussi</i> étiquette de sous-titre.
étiquette de sous-titre	Etiquette sur un objet de face-avant utilisée pour donner un nom à l'objet dans l'interface utilisateur. Ce nom peut être traduit dans d'autres langages sans affecter le code source du diagramme. <i>Voir aussi</i> Etiquette de nom.
exécution asynchrone	Mode dans lequel plusieurs processus se partagent le temps du processeur. Par exemple, un processus s'exécute pendant que d'autres attendent des interruptions au cours d'une E/S de périphérique ou pendant qu'elles attendent une impulsion d'horloge.

exécution en continu	Mode d'exécution dans lequel un VI est exécuté indéfiniment jusqu'à ce que l'opérateur l'arrête. Vous l'activez en cliquant sur le bouton Exécution en continu .
exécution intégrée	Aptitude d'une fonction ou d'un VI à réutiliser de la mémoire au lieu d'en allouer davantage.
exécution pilotée par tableau	Méthode d'exécution dans laquelle les tâches individuelles sont des conditions séparées dans une structure Condition qui est imbriquée dans une boucle While. Les séquences sont spécifiées sous forme de tableaux de numéros de conditions.
exécution ré-entrante	Mode dans lequel des appels à plusieurs instances d'un sous-VI peuvent s'exécuter en parallèle avec un stockage de données distinct et séparé.
exécution en mode Animation	Fonction qui anime l'exécution du VI pour illustrer le flux de données dans le VI.

F

face-avant	Interface utilisateur interactive d'un VI. Modélisée à partir de la face-avant d'instruments physiques, elle est composée d'interrupteurs, de glissières, de vu-mètres, de graphes, de graphes déroulants, de jauges, de LED et d'autres commandes et indicateurs.
fenêtre active	Fenêtre qui est actuellement configurée pour accepter les entrées de l'utilisateur, généralement la fenêtre qui se trouve au premier plan. La barre de titre d'une fenêtre active est en surbrillance. Vous rendez une fenêtre active en cliquant dessus, ou en la sélectionnant dans le menu Fenêtres .
fenêtre d'aide	Fenêtre spéciale qui affiche les noms et emplacements des terminaux d'une fonction ou d'un sous-VI, la description des commandes et indicateurs, les valeurs des constantes universelles et les descriptions et les types de données des attributs des commandes. La fenêtre donne également accès à la Référence en ligne pour le G.
fenêtre Hiérarchie	Fenêtre qui affiche graphiquement la hiérarchie des VIs et des sous-VIs.
FFT	Fast Fourier transform (Transformée de Fourier rapide).

fichier d'enregistrement de données	Fichier qui stocke des données sous forme d'une séquence d'enregistrements d'un type de données arbitraire unique que vous devez spécifier lorsque vous créez le fichier. Alors que tous les enregistrements d'un fichier d'enregistrement de données doivent être du même type, ce type peut être complexe ; par exemple, vous pouvez spécifier que chaque enregistrement est un cluster contenant une chaîne de caractères, un nombre et un tableau.
fichier des chaînes de caractères du VI	Fichier texte muni de tags, contenant toutes les chaînes de caractères de localisation qui se trouvent sur la face-avant d'un VI.
fichier de structure d'octet	Fichier qui stocke des données sous forme de séquence de caractères ASCII ou d'octets.
fil de liaison	Chemin de données entre nœuds.
flux de données	Système de programmation consistant en des nœuds exécutables, dans lequel les nœuds s'exécutent uniquement lorsqu'ils ont reçu toutes les entrées de données requises et produisent des sorties automatiquement lorsque leur exécution est terminée. Le G est un système à flux de données.
fonction	Élément d'exécution intégré, comparable à un opérateur, une fonction ou une instruction dans un langage conventionnel.
formats de stockage de données	Organisation et représentation des données stockées en mémoire.

G

G	Langage de programmation graphique utilisé dans les applications LabVIEW et BridgeVIEW.
glisser	Glisser le curseur de la souris sur l'écran pour sélectionner, déplacer, copier ou supprimer des objets.
glissière	Partie mobile des commandes et indicateurs à glissière.
glyphe	Petite image ou icône.
GPIB	General Purpose Interface Bus (bus d'interface universel). Nom courant du système d'interface de communications défini dans la norme ANSI/IEEE 488.1-1987 et dans la norme ANSI/IEEE 488.2-1987. Hewlett-Packard, l'inventeur du bus, l'appelle HP-IB.

graphe à balayage	Semblable à un oscillographe, à ceci près qu'une ligne se déplace sur l'affichage pour séparer les anciennes données des nouvelles.
graphe déroulant	Indicateur de tracé numérique modélisé sur un enregistreur de graphe déroulant à ruban de papier, qui se déroule à mesure qu'il trace les données. <i>Voir aussi</i> oscillographe, graphe à balayage et graphe oscilloscopique.
graphe oscilloscopique	Indicateur qui trace des points de données à une certaine vitesse.

H

handle	Pointeur vers un pointeur de bloc de mémoire ; les handles font référence à des tableaux et à des chaînes de caractères. Un tableau de chaînes de caractères est un handle sur un bloc de mémoire contenant des handles de chaînes de caractères.
hex	Hexadécimal. Un système de nombres en base 16.
Hz	Hertz. Cycles par seconde.

I

icône	Représentation graphique d'un nœud sur un diagramme.
IEEE	Institute for Electrical and Electronic Engineers (Institut pour ingénieurs électriciens et électroniciens).
impression contrôlée par programme	Impression automatique de la face-avant d'un VI après exécution.
indicateur	Objet de la face-avant qui affiche une sortie.
Inf	Valeur d'affichage numérique pour une représentation à virgule flottante de l'infini.
info-bulle	Petite zone de texte qui affiche le nom d'un objet, d'une commande ou d'un terminal.
instrument virtuel	Programme dans LabVIEW ou BridgeVIEW ; ainsi appelé parce qu'il reproduit l'apparence et la fonction d'un instrument physique.

L

LabVIEW	Laboratory Virtual Instrument Engineering Workbench. Application de développement de programme basée sur le langage de programmation G utilisée couramment pour des mesures et des tests.
LED	Diode électroluminescente.
légende	Objet appartenant à un graphe ou à un graphe déroulant, qui affiche le nom et le style des tracés qui figurent sur ce graphe ou graphe déroulant.

M

marquise	Cadre mobile, en pointillés, qui entoure des objets sélectionnés.
matrice	Tableau à deux dimensions.
menus déroulants	Menus accessibles à partir d'une barre de menus. Les options des menus déroulants sont souvent de nature générale.
menu local	Menu auquel on accède en cliquant sur un objet avec le bouton droit de la souris (Windows) ou en maintenant enfoncée la touche de commande (Macintosh).
mise à l'échelle automatique	Aptitude des échelles à s'ajuster à la gamme de valeurs tracées. Sur les échelles des graphes, cette fonction détermine la valeur maximale et la valeur minimale de l'échelle.
mnémonique	Chaîne de caractères associée à une valeur entière.
Mo	Méga-octets de mémoire.
multithreading	Une application multithread est une application qui exécute indépendamment plusieurs threads d'exécution différents. Sur un ordinateur à multiples processeurs, les différents threads peuvent s'exécuter simultanément sur différents processeurs.

N

NaN	Valeur d’affichage numérique pour une représentation à virgule flottante de <i>not a number (pas un nombre)</i> . Il s’agit généralement du résultat d’une opération non définie, telle que $\log(-1)$.
nœuds	Éléments d’exécution d’un diagramme comprenant des fonctions, des structures et des sous-VIs.
nœud Assembler	Fonction qui crée des clusters à partir de différents types d’éléments.

O

objet	Terme générique pour n’importe quel élément de la face-avant ou du diagramme, y compris les commandes, les nœuds, les fils de liaison et les images importées.
oscillographe	Indicateur numérique modélisé sur le fonctionnement d’un oscilloscope.
outil	Curseur spécial que vous pouvez utiliser pour effectuer des opérations spécifiques.
outil Bobine	Outil utilisé pour définir des chemins de données entre les terminaux source et destination.
outil Doigt	Outil utilisé pour entrer des données dans des commandes et pour les exécuter. Ressemble à un doigt tendu.
outil Flèche	Outil ressemblant à une flèche et utilisé pour déplacer, sélectionner et redimensionner des objets.
outil Main	Outil utilisé pour faire défiler les fenêtres.
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 des couleurs afin de les coller avec l’outil Pinceau.
outil Point d’arrêt	Outil utilisé pour placer un point d’arrêt sur un VI, un nœud ou un fil de liaison.
outil Sonde	Outil utilisé pour placer des sondes sur des fils de liaison.

outil Texte	Outil utilisé pour créer des étiquettes et entrer du texte dans des fenêtres de texte.
ouvrir un menu local	Invoquer un menu spécial en cliquant sur un objet avec le bouton droit de la souris (Windows) ou en maintenant enfoncée la touche de commande (Macintosh).

P

palette	Affichage des icônes qui représentent les options possibles.
palette de Commandes	Palette contenant des commandes et indicateurs de face-avant.
palette de Fonctions	Palette contenant des structures de diagramme, des constantes, des fonctions de communication et des VIs.
palette d'outils	Palette contenant des outils que vous pouvez utiliser pour éditer et mettre au point les objets de la face-avant et du diagramme.
palette hiérarchique	Menu qui contient des palettes et des sous-palettes.
pas un chemin	Valeur prédéfinie pour la commande de chemin, qui signifie que le chemin n'est pas valide.
pas un refnum	Valeur prédéfinie qui signifie que le refnum n'est pas valide.
pixmap	Format standard pour stocker des images, dans lequel chaque pixel est représenté par une valeur de couleur. Une bitmap est une version noir et blanc d'une pixmap.
plate-forme	Ordinateur et son système d'exploitation.
point d'arrêt	Utilisé pour interrompre l'exécution lors de l'appel à un sous-VI. Vous pouvez placer un point d'arrêt en cliquant sur un VI, un nœud ou un fil de liaison avec l'outil Point d'arrêt qui se trouve dans la palette Outils .
point de coercion	Glyphe sur un terminal indiquant qu'un ou deux terminaux câblés ensemble ont été convertis pour correspondre à leur type de données respectif.
polymorphisme	Aptitude d'un nœud à s'ajuster automatiquement à des données de représentation, de type ou de structure différents.

programmation modulaire	Programmation qui utilise des programmes informatiques interchangeables.
programmation séquentielle	Système de programmation dans lequel l'ordre séquentiel des instructions détermine l'ordre d'exécution. La plupart des langages de programmation textuels conventionnels, tels que le C, le Pascal et le BASIC, sont des langages de programmation séquentielle.
pseudo-code	Représentation simplifiée, indépendante du langage, de code de programmation.

R

refnum	Identificateur que le G associe à un fichier lorsque vous l'ouvrez. Vous devez utiliser le refnum de fichier pour indiquer que vous souhaitez qu'une fonction ou un VI réalise une opération sur le fichier qui est ouvert.
registre à décalage	Mécanisme optionnel utilisé dans des structures de boucle pour transmettre la valeur d'une variable d'une itération de la boucle à l'itération suivante.
représentation	Sous-type du type de données numérique, qui comprend des entiers octets, mots et longs signés et non signés, ainsi que des nombres à virgule flottante en simple précision, en double précision et en précision étendue, à la fois réels et complexes.
routine externe partagée	Sous-programme qui peut être partagé par plusieurs ressources de code CIN.

S

scalaire	Nombre pouvant être représenté par un point sur une échelle. Une valeur unique, par opposition à un tableau. Les booléens et clusters scalaires sont des instances explicitement singulières de leurs types de données respectifs.
sonde	Fonction de mise au point permettant de vérifier des valeurs intermédiaires dans un VI.
sous-diagramme	Diagramme à l'intérieur des limites d'une structure.
sous-VI	VI utilisé dans le diagramme d'un autre VI ; comparable à un sous-programme.

squelette de cluster	Objet de la face-avant qui contient les éléments d'un cluster.
structure	Élément de commande de programme, tel qu'une structure Séquence, une structure Condition, une boucle For ou une boucle While.
structure Condition	Structure de commande à branchement conditionnel, qui exécute un et un seul de ses sous-diagrammes, en fonction de son entrée. Elle est similaire aux instructions IF-THEN-ELSE et CASE des langages de programmation séquentielle.
structure Séquence	Structure de commande de programme qui exécute ses sous-diagrammes par ordre numérique. Couramment utilisée pour forcer des nœuds qui ne dépendent pas de données à s'exécuter dans un ordre désiré.

T

tableau	Liste ordonnée et indexée d'éléments de données du même type.
tableau vide	Tableau contenant zéro élément, mais qui a un type de données défini. Par exemple, un tableau qui a une commande numérique dans sa fenêtre d'affichage de données mais n'a de valeur définie pour aucun élément est un tableau numérique vide.
tableau vierge	Objet de la face-avant qui accueille un tableau. Il comporte un affichage de l'indice, une fenêtre d'objet de données et une étiquette optionnelle. Il peut accepter divers types de données.
terminal	Objet ou zone d'un nœud au travers duquel sont transmises des données.
terminal conditionnel	Terminal d'une boucle While contenant une valeur booléenne qui détermine si le VI effectue une autre itération.
terminal de comptage	Terminal d'une boucle For dont la valeur détermine le nombre de fois que la boucle For doit exécuter son sous-diagramme.
terminal de destination	Terminal qui reçoit des données.
terminal d'itération	Terminal d'une boucle For ou d'une boucle While qui contient le nombre d'itération actuellement effectuées.
terminal source	Terminal qui émet des données.
texte libre	Étiquette sur la face-avant ou le diagramme qui n'appartient à aucun autre objet.

tracé Représentation graphique d'un tableau de données, affichée soit sur un graphe soit sur un graphe déroulant.

tunnel Terminal d'entrée ou de sortie de données sur une structure.

U

UUT Unit Under Test (Unité en cours de test).

V

variable locale de séquence Terminal qui transmet des données entre les étapes d'une structure Séquence.

VI *Voir* instrument virtuel.

VI à souche Prototype non fonctionnel d'un sous-VI. Il a des entrées et des sorties, mais est incomplet. Il est utilisé lors des premiers stades de l'élaboration d'un VI pour réserver des emplacements pour le développement ultérieur du VI.

VI actuel VI dont la face-avant, le diagramme ou l'Editeur d'icône est la fenêtre active.

VI invalide VI qui ne peut pas être compilé ni exécuté ; désigné par une flèche brisée dans le bouton Exécution.

VI principal VI situé en haut de la hiérarchie des VIs. Ce terme distingue le VI de ses sous-VIs.

VI Serveur Mécanisme pour contrôler les VIs et les applications en G au moyen d'un programme. Peut également être utilisé pour commander à distance les VIs ou les applications en G.

Z

zone de liste	Boîte à l'intérieur d'une boîte de dialogue listant tous les choix disponibles pour une commande, comme par exemple une liste de noms de fichiers sur un disque. En général, vous sélectionnez un élément dans la zone de liste, puis vous cliquez sur OK . Si la zone de liste ne peut pas afficher tous les choix existants, elle dispose de barres de défilement. En sélectionnant la flèche vers le bas à côté du premier élément dans la liste, vous affichez le reste de la zone de liste.
zone de redimensionnement	Zone triangulaire située aux coins d'un objet et indiquant des points de redimensionnement.

Index

Symboles

- <Alt-b>, supprimer les fils incorrects, 4-9
- <bouton du milieu-clic>, pour décrocher la dernière fixation, 18-3
- <commande>-clic pour accéder aux menus locaux, 1-4
- <Ctrl-b>, supprimer les fils incorrects, 4-9
- <Ctrl-clic>, pour décrocher la dernière fixation, 18-3
- <Maj>-clic sur les objets, 2-7
- <meta-b>, supprimer les fils incorrects, 4-9
- <option-clic>, pour décrocher la dernière fixation, 18-3

A

- Abandonner, bouton. *Voir* Stop, bouton
- action Armement au relâchement, 10-5, 10-6, 24-14
- action Armement jusqu'au relâchement, 10-6
- action Commutation à l'appui, 10-5
- action Commutation au relâchement, 10-5, 24-14
- action Commutation jusqu'au relâchement, 10-5
- Activer l'enregistrement/l'impression après l'exécution, option, 6-5
- affichage de l'indice des tableaux
 - affichage de l'indice sous forme élément unique ou tabulaire, 14-10 à 14-11
 - interprétation, 14-9
 - menu local, 14-7
 - tableau vierge, 14-6
- Afficheur numérique, option, 2-14
- aide
 - Attribute Nodes, 1-9
 - fonction et utilisation, 1-7
 - option Aide simple, 1-8
 - option Référence en ligne, 1-10
 - option Verrouiller l'aide, 1-7
 - option Visualiser l'aide, 1-7
 - touche <aide>, 1-7
- Ajouter des fonctions
 - icône, 1-5
- Ajuster à la taille de l'écran, option, 6-5
- Alignement, menu, 2-11
- appel de code d'autres langages, 25-1 à 25-14
 - Code Interface Nodes, 25-3
 - exécution d'autres applications de l'intérieur des VIs, 25-1 à 25-3
 - utilisation du convertisseur de panneau de fonction LabWindows/CVI, 25-9 à 25-14
 - VIs AppleEvent, 25-1
- applications, gestion
 - développeurs multiples, 27-8 à 27-13
 - distribution des VIs, 27-2 à 27-7
 - enregistrer des VIs dans des bibliothèques, 27-1 à 27-2
 - historique, tenue à jour
 - fenêtre Historique, 27-9 à 27-13
 - impression des informations de l'historique, 27-13
 - numéros de révision, 27-11 à 27-12
 - options associées des boîtes de dialogue
 - Configuration du VI et Préférences, 27-13
 - réinitialisation des informations de l'historique, 27-12
 - sauvegarde de fichiers, 27-2
 - stockage de VIs dans des bibliothèques, 2-30
 - utiliser Enregistrer avec options, 27-3
- arrêt des VIs, 4-4
- Associer une touche, boîte de dialogue, 8-6 à 8-8

- Associer une touche, option
 - boîte de dialogue (illustration), 8-7
 - fonction et utilisation, 8-6
 - attribut "Focus de touche", 22-7
 - attribut Clignotement, 22-8
 - attribut Couleur du tracé, 22-10
 - attribut Courbe active, 22-9 à 22-10
 - attribut Désactiver, 22-7
 - attribut Double-clic, 22-12
 - attribut Info de l'échelle des Z
 - Couleur inférieure, 15-38
 - Couleur supérieure, 15-38
 - Tableau de couleurs, 15-38
 - attribut Limites (lecture seule), 22-9
 - attribut Position, 22-8
 - attribut Table des couleurs, 15-38
 - attribut Visible, 22-6
 - Attribute Nodes, 22-1 à 22-14
 - aide, 1-9
 - attributs de base, 22-6 à 22-12
 - attribut "Focus de touche", 22-7
 - attribut Clignotement, 22-8
 - attribut Couleur du tracé, 22-9 à 22-10
 - attribut Désactiver, 22-7
 - attribut Double-clic, 22-12
 - attribut Limites (lecture seule), 22-9
 - attribut Position, 22-8
 - courbe active, 22-9 à 22-10
 - création, 22-1 à 22-6
 - associer un terminal à l'attribut node, 22-6
 - lecture ou définition des attributs, 22-3
 - plusieurs noeuds, 22-4
 - exemples, 22-9
 - booléens, 22-10
 - définition des chaînes de caractères des commandes de type roue codeuse, 22-11
 - lecture des curseurs par programme, 22-13
 - présentation des options aux utilisateurs, 22-13
 - menu local, 22-1 à 22-2
 - recherche, 22-6
 - auto-indexation
 - boucles While, 19-8
 - définition, 19-7
 - insuffisance de mémoire (remarque), 19-9
 - intérêt et utilisation, 19-7
 - nombre de la boucle For, 19-8
 - Autoriser les menus locaux pendant l'exécution, case, 6-4
 - avertissements
 - affichage, préférences, 7-14
- ## B
- barre d'outils
 - masquée, 6-5
 - barre de menu
 - problèmes causés par le masquage (avertissement), 6-5
 - barre de menus
 - définition, 2-6
 - préférences des couleurs, 7-14
 - barres de défilement
 - préférences des couleurs, 7-15
 - utilisation avec des commandes et indicateurs de chaîne de caractères, 11-3
 - Basculement horizontal, option, 3-6
 - Basculement vertical, option, 3-6
 - bases pour entiers, sélection, 9-4
 - bibliothèque
 - répertoire
 - spécification du chemin, 7-4
 - bibliothèques de VIs, 2-30
 - arranger des fichiers dans, 27-1 à 27-2
 - création, 2-31 à 2-32

- enregistrement de fichiers, 2-30
- enregistrement de VIs dans le répertoire vi.lib (avertissement), 2-30
- suppression de VIs, 2-32
- bibliothèques. *Voir* bibliothèques de VIs
- blocages du système, questions concernant, B-5
- Bobine, outil
 - affectation de terminaux, 3-7 à 3-8
 - illustration, 2-5
- Boîte de calcul, 20-1 à 20-10
 - commentaires, 20-1
 - définition, 20-1
 - erreurs (tableau), 20-9
 - erreurs de syntaxe, 20-4
 - menu local, 20-2
 - noms de fonctions (tableau), 20-4 à 20-7
 - notation Backus-Naur Form, 20-7
 - scalaires numériques à virgule flottante, 20-3
 - syntaxe, 20-1, 20-2, 20-7 à 20-8
 - variables d'entrée et de sortie, 20-2 à 20-3
- boîte de couleurs
 - commandes et indicateurs numériques, 9-23 à 9-24
 - illustration, 9-23
- boîte de dialogue
 - Configuration du noeud du sous-VI, 6-1
 - Configuration du VI, 6-1
- boîte de dialogue Gamme des données, 9-6
- boîte de dialogue Imprimer la documentation. *Voir aussi* boîte de dialogue Paramètres d'impression personnalisée
 - bouton Configurer, 5-4
 - choix d'options de mise en page
 - Ajuster la taille de la face-avant, 5-5
 - Ajuster la taille du diagramme, 5-5
 - Imprimer les en-têtes des sections, 5-5
 - Imprimer un en-tête, 5-5
- définition des formats d'impression
 - format Documentation complète, 5-4
 - format Icône, description, face-avant et diagramme, 5-4
 - format Personnalisé, 5-4
 - format Utilisation comme sous-VI, 5-4
 - format Utilisation de la face-avant, 5-4
- illustration, 5-3
- impression/exportation vers des fichiers RTF ou HTML, 5-10 à 5-15
- Sauts de page entre sections, 5-5
- boîte de dialogue Paramètres d'impression personnalisée, 5-6
 - Commandes, 5-7
 - Diagramme, 5-7
 - Face-avant, 5-7
 - Hierarchie du VI, 5-7
 - Historique du VI, 5-7
 - Icône et description, 5-6
 - illustration, 5-6
 - Liste des sous-VIs, 5-7
- Boîte de dialogue, option, 6-4
- Booléen, palette, 8-3
- boucle For
 - auto-indexation, 19-7 à 19-8
 - éviter les calculs superflus, 28-11 à 28-13
 - exécution des types zéro, 19-9
 - icône, 19-3
 - intérêt et utilisation, 17-11, 19-4
 - placement d'objets à l'intérieur des structures, 19-4
 - registres à décalage, 19-9 à 19-12
 - terminaux à l'intérieur des boucles, 19-6
- boucle While
 - auto-indexation, 19-7 à 19-8
 - éviter les calculs superflus, 28-11 à 28-13
 - icône, 19-4
 - intérêt et utilisation, 19-4

- placement d'objets à l'intérieur des structures, 19-4
- registres à décalage, 19-9 à 19-12
- terminaux à l'intérieur des boucles, 19-6
- boucles For
 - intérêt et utilisation, 17-11
- boucles While
 - but et utilisation, 17-11
- boucles. *Voir* boucle For; boucle While
- bouton Abandonner
 - cache, 4-4
 - intérêt et utilisation, 4-1, 4-4
- bouton Commande invalide, 24-4
- bouton droit de la souris pour accéder aux menus locaux, 1-4
- bouton Effacer, enregistrement des données de la face-avant, 4-6
- bouton Entrée, 14-23
 - changer les valeurs des commandes et indicateurs numériques, 9-2
 - enregistrement des données de la face-avant, 4-6
- bouton Exécuter
 - exécution des VIs, 4-1
 - utilisation pendant une interruption de l'exécution, 4-31
 - VI appelant en cours d'exécution, 4-1
 - VI s'exécutant au niveau principal, 4-1
- bouton Exécuter en continu, 4-1
- bouton Exécution détaillée, 4-20 à 4-21
- bouton Exécution en mode Animation, 4-22
- bouton Exécution non détaillée, 4-20 à 4-21
- bouton Exécution permanente, 4-4
- bouton Exécution, brisé, 4-9
- bouton Flèche brisée, 4-10, 18-8
- bouton Mise en garde, accès à la fenêtre Liste des erreurs, 4-10
- bouton Parcourir, 16-4
- bouton Pause, 4-2, 4-19
- bouton Retour au début, 4-31
- bouton Revenir au VI appelant, 4-31

- bouton Sortie, 4-20 à 4-21
- bouton X, 14-23
- branche (de fils), 18-6
- bulles de données pendant l'exécution, sélection, 7-14

C

- câblage des diagrammes, 18-1 à 19-26
 - adaptation des liaisons, 18-5 à 18-6
 - insertion d'objets, 17-12
 - points de fixation, 18-3
 - problèmes de câblage
 - absence de source de liaison, 18-11
 - conflit d'élément, 18-9
 - conflit d'unités, 18-9
 - conflit de dimension, 18-9
 - conflit de type, 18-9
 - cycle du fil de liaison, 18-12
 - liste des erreurs, 18-8
 - mauvaises connexions de fils, 18-11
 - plusieurs sources de fil de liaison, 18-10
 - segments déconnectés, 18-11
- problèmes de câblage de structure
 - affectation de plusieurs valeurs à, 19-22
 - câblage à partir de plusieurs cadres d'une structure Séquence, 19-24
 - câblage par dessous plutôt qu'à travers des structures, 19-25
 - chevauchement de tunnels, 19-23
- remplacement d'objets, 17-12
- situations à éviter
 - câblage sous des objets, 18-14
 - fils de liaison en boucle, 18-13
 - portions du fil masquées, 18-13

- symbole de la souris de l'outil
 - Bobine, 18-1
- techniques de base du câblage, 18-1 à 18-14
- VI's compliqués, 18-4 à 18-5
- cache
 - bouton Abandonner, 4-4
- cadre connecteur
 - Cette connexion est, option, 3-9
 - option Basculement horizontal, 3-6
 - option Basculement vertical, 3-6
 - option Déconnecter ce terminal, 3-10
 - option Déconnecter tous les terminaux, 3-10
 - option Rotation de 90 Degrés, 3-6
- cadre de sélection, 2-7
- cadre icône
 - menu local, 3-5
- cadre, impression autour de la face-avant, 5-10
- cadres, dans les structures Séquence, 19-18
- caractères non affichables, insertion avec des codes à barre oblique inverse (\), 11-2, 11-4 à 11-6
 - exemples, 11-4 à 11-6
 - table des codes, 11-4
- Centrage automatique, option, 6-5
- Chaîne, palette, 8-3
- chaînes d'appel, lecture, 4-21
- chaînes de caractères
 - données aplaties, A-14
 - formats d'enregistrement des données, A-5
- Changer en commande, option, 8-3
- Changer en indicateur, option, 8-3
- charger dynamiquement des VI's, B-12
- chemin
 - Chemin du recherche du VI, 7-4 à 7-6
 - données aplaties, A-14
 - formats d'enregistrement des données, A-5
- préférences
 - boîte de dialogue, 7-2
 - répertoire de bibliothèque, 7-3
 - répertoire par défaut, 7-4
 - répertoire temporaire, 7-4
- Chemin du recherche du VI, 7-4 à 7-6
- CIN (Code Interface Nodes), 25-3, 26-6, 26-7
 - exécution synchrone, 26-7
- clignotement
 - préférences de vitesse, 7-11
- clonage d'objets, 2-13
- clusters, 14-20 à 14-33
 - assemblage
 - fonction "Assembler par nom", 14-26 à 14-28
 - fonction "Tableau en cluster", 14-2, 14-28
 - fonction Assembler, 14-25
 - comparés aux tableaux, 14-20
 - configuration des éléments du cluster, 14-22 à 14-24
 - création, 14-21
 - définition, 14-20
 - définition de type, 24-23
 - déplacement, 14-24
 - désassemblage
 - fonction "Cluster en tableau", 14-32
 - fonction "Séparer par nom", 14-30 à 14-32
 - fonction Séparer, 14-30
 - désassemblage des éléments du cluster, 14-20
 - descripteurs de type, A-12
 - dimensionnement, 14-24
 - données aplaties, A-15
 - exécution des éléments du cluster, 14-22 à 14-24
 - formats d'enregistrement des données, A-5 à A-7
 - intérêt, 14-20
 - motifs de fil, 14-20

- ordonnancement des éléments du cluster, définition, 14-23
- ordre du cluster, A-5 à A-7
- points, 14-20
- redimensionnement, 14-24
- remplacement d'éléments du cluster, 14-33
- valeurs par défaut, définition, 14-22
- codes à barre oblique inverse (\)
 - entrée et visualisation de caractères non affichables, 11-2
 - exemples, 11-6
 - table des codes, 11-4
- Coin, outil, 2-23
- Coller, 2-13
- colonnes, redimensionnement, 11-8
- commande Ajouter, 17-10
- commande Ajouter un intervalle entre les éléments, 14-15
- commande Ajouter une dimension, 14-8
- commande de refnum de VI de type strict, 21-10 à 21-12
 - à quel moment l'utiliser, 21-10
 - création, 21-11
 - exemple, 21-11
 - exigée avec la fonction "Ouvrir une référence de VI", 12-5, 21-10
- commande Dimensionnement automatique, 14-24
- commande et indicateur de variable ActiveX, 16-1 à 16-3
- commande Exécution, 4-1
- commande Importer une image, 24-11
- commande Liste des erreurs, 4-10
- commande Prendre la valeur actuelle par défaut, 14-22
- commande Stop, 4-4
- commande Supprimer, 17-10
- commande Vider le tableau, 14-13
- commande Visualiser base, 9-4
- <commande-M> raccourci pour passer entre édition et exécution (avertissement), 6-5
- commandes
 - affectation de terminaux, 3-7
 - Associer une touche, boîte de dialogue, 8-6 à 8-8
 - boîte de dialogue, 8-9 à 8-10
 - changer en indicateurs, 8-3
 - commandes en composantes, 24-16 à 24-17
 - couper, copier et coller, 8-3
 - définition, 1-3
 - masquer, 2-14
 - options communes, 8-2 à 8-6
 - ordre des panneaux, 8-8 à 8-9
 - remplacement, 8-4 à 8-6
 - suppression interdite, 2-14
- commandes de boîte de dialogue, 8-9 à 8-10
- commandes de refnum, 12-3 à 12-7
 - commande de refnum de VI de type strict, 12-5
 - exemples, 12-6 à 12-7
 - intérêt et utilisation, 12-3
 - RefNum d'occurrence, 12-4
 - RefNum de connexion réseau, 12-5
 - RefNum de fichier d'enregistrement de données, 12-4
 - RefNum de fichier de structure d'octet, 12-4
 - RefNum de file d'attente, 12-6
 - RefNum de notificateur, 12-5
 - RefNum de périphérique, 12-4
 - RefNum de rendez-vous, 12-6
 - RefNum de sémaphore, 12-6
 - Refnum de VI ou d'application, 12-5
 - RefNum VISA, 12-5
 - types (figure), 12-3
- commandes de tableau, 14-2 à 14-20
 - affichage de l'indice
 - interprétation, 14-9
 - menu local, 14-7

- affichage de tableaux sous forme élément unique ou tabulaire, 14-10 à 14-11
- création, 14-5 à 14-11
 - définition du type de tableau, 14-6
 - menu local de l’affichage de l’indice, 14-7
 - définition de la taille du tableau, 14-8
- commandes de type énuméré, 13-13 à 13-14
 - illustration, 13-13
 - intérêt et usage, 13-14
 - par rapport aux commandes de menu déroulant, 13-14
- commandes de type roue codeuse
 - exemple d’Attribute Node, 22-9
- commandes en composantes, 24-16 à 24-17
- commandes et indicateurs booléens, 10-1 à 10-6
 - configuration, 10-3 à 10-6
 - configuration de l’action mécanique, 10-4 à 10-6
 - contrôle de la gamme de données, 10-4
 - création, 10-1 à 10-2
 - étiquettes, 10-3 à 10-4
 - exemple d’Attribute Node, 22-11
 - illustration, 10-2
 - options du menu local, 10-3
 - Libérer le texte, 10-3
 - personnalisation à l’aide d’images importées, 10-6
 - utilisation, 10-1 à 10-2
- commandes et indicateurs de chaîne de caractères, 11-1 à 11-6
 - aperçu, 11-1 à 11-2
 - entrer des caractères non affichables, 11-2
 - option Affichage hexadécimal, 11-6
 - option Affichage style mot de passe, 11-6
 - option Limiter à une seule ligne, 11-6
 - options, 11-3 à 11-6
 - utilisation de la barre de défilement, 11-3
- commandes et indicateurs de chemin, 12-1 à 12-2
 - illustration, 12-1
 - intérêt et utilisation, 12-1
 - symbole Chemin, 12-2
 - symbole Pas un chemin, 12-2
- commandes et indicateurs numériques, 9-1 à 9-11
 - afficher des entiers dans d’autres bases, 9-4
 - boîte de couleurs, 9-23 à 9-24
 - bouton Entrée pour remplacer les anciennes valeurs, 9-2
 - changer la représentation des valeurs numériques, 9-5
 - commandes et indicateurs numériques, 9-1 à 9-11
 - afficher des entiers dans d’autres bases, 9-4
 - bouton Entrée pour remplacer les anciennes valeurs, 9-2
 - changer la représentation des valeurs numériques, 9-5
 - contrôle des gammes numériques, 9-7 à 9-9
 - format et précision des affichages numériques, 9-9 à 9-11
 - illustration, 9-1
 - incréméntation et décrémentation, 9-2 à 9-3
 - intérêt et utilisation, 9-2 à 9-3
 - options de gamme, 9-6
 - options du menu local d’une valeur numérique, 9-4
 - outil Doigt, 9-2
 - valeurs acceptables, 9-2
- commandes et indicateurs numériques à glissière, 9-12
 - échelle de la glissière, 9-14 à 9-19
 - échelle du texte, 9-18 à 9-19
 - glissières pleines et à valeurs multiples, 9-19

- illustration, 9-12
- menu local d'échelle, 9-14 à 9-15
- options du menu local de glissière, 9-14
- présentation, 9-13
- repères d'échelle, 9-15 à 9-19
- utilisation des glissières, 9-13
- commandes et indicateurs rotatifs, 9-21 à 9-23
 - illustration, 9-21
 - utilisation et modification, 9-21 à 9-23
- contrôle des gammes numériques, 9-7 à 9-9
- format et précision des affichages numériques, 9-9 à 9-11
 - boîte de dialogue Format & Précision, 9-10
 - exemples, 9-10
 - temps absolu et date, 9-10 à 9-11
- illustration, 9-1
- incrémentement et décrémentation, 9-2 à 9-3
- intérêt et utilisation, 9-2 à 9-3
- options de gamme, 9-6
 - boîte de dialogue Gamme des données, 9-6
 - contrôle des gammes numériques, 9-7 à 9-9
 - gamme des nombres à virgule flottante en précision étendue, 9-6
 - nombres à virgule flottante (tableau), 9-6
- options du menu local d'une valeur numérique, 9-4
- outil Doigt, 9-2
- palette (illustration), 2-2
- rampe de couleurs, 9-24 à 9-25
- types d'unités, 9-25 à 9-33
 - autres unités (tableau), 9-29
 - entrée des unités, 9-30
 - unités CGS (tableau), 9-29
 - unités de base (tableau), 9-27
 - unités dérivées à noms spéciaux (tableau), 9-27 à 9-28
 - unités et contrôle de type strict, 9-31 à 9-32
 - unités polymorphes, 9-32
 - unités supplémentaires en usage avec des unités SI (tableau), 9-28
 - visualisation de l'étiquette d'unité, 9-25
 - valeurs acceptables, 9-2
- commandes et indicateurs numériques à glissière
 - échelle de la glissière, 9-14 à 9-19
 - échelle du texte, 9-18 à 9-19
 - glissières pleines et à valeurs multiples, 9-19
 - illustration, 9-12
 - menu local d'échelle, 9-14 à 9-15
 - options du menu local de glissière, 9-14
 - présentation, 9-13
 - repères d'échelle, 9-15 à 9-19
 - changer les limites d'une échelle, 9-15 à 9-16
 - déplacer des repères arbitraires, 9-17
 - mode Repères arbitraires, 9-17
 - mode Uniforme, 9-17
 - sélectionner une distribution non uniforme de repères d'échelle, 9-16 à 9-18
 - supprimer un repère arbitraire, 9-17
 - utilisation des glissières, 9-13
- commandes et indicateurs numériques rotatifs, 9-21 à 9-23
 - illustration, 9-21
 - utilisation et modification, 9-21 à 9-23

- commandes et indicateurs. *Voir aussi* types spécifiques
 - création automatique, 17-13
 - terminaux, 17-1 à 17-3
 - tableau de symboles, 17-3
- commandes personnalisées
 - ajout à la palette Commandes, 24-1
 - application des modifications des commandes personnalisées, 24-4
 - avertissements, 24-18 à 24-19
 - commandes en composantes, 24-16 à 24-17
 - composante en cours, 24-8
 - composantes de texte, 24-15
 - composantes décoratives
 - images indépendantes, 24-14 à 24-15
 - plusieurs images, 24-12
 - composantes indépendantes, 24-7 à 24-8
 - création, 24-6
 - création d'icônes, 24-5
 - éditeur de commandes, 24-2
 - enregistrement, 24-5
 - fenêtre des composantes de l'éditeur de commandes, 24-8 à 24-9
 - indépendance du fichier source, 24-6
 - menus locaux des différentes composantes, 24-10
 - ouverture, 24-6
 - utilisation, 24-5
 - valider les commandes personnalisées, 24-4
- Commandes, palette
 - options, 8-1
 - sous-menu Visualiser, 8-3
- comment construire des palettes
 - ActiveX, 16-5
- Compacter la mémoire pendant l'exécution, option, 7-9
- complexe double précision (CDB), 9-5
- complexe précision étendue (CXT), 9-5
- complexe simple précision (CSG), 9-5
- composantes de texte, 24-15
- composantes décoratives
 - copie des composantes ou des décorations de LabVIEW, 24-11
 - images indépendantes, 24-14 à 24-15
 - menu local, 24-10
 - plusieurs images, 24-12
- composantes des commandes
 - commandes en composantes, 24-16 à 24-17
 - composantes de texte, 24-15
 - composantes décoratives, 24-10 à 24-15
 - ajout aux commandes personnalisées, 24-17 à 24-18
 - avec des images indépendantes, 24-14 à 24-15
 - avec plusieurs images, 24-12
- composantes indépendantes, 24-7 à 24-8
- conditions de compétition, 26-16 à 26-17
- configuration de LabVIEW. *Voir* préférences
- Configuration du noeud du sous-VI, options, 6-7
- Configuration du VI, options
 - création de panneaux locaux, 6-1 à 6-2
 - documentation, 6-6 à 6-7
 - fenêtres, 6-3 à 6-5
 - options d'exécution, 6-2 à 6-3
- connecteurs
 - accès avec l'option Visualiser le connecteur, 1-6
 - comparaison avec les paramètres, 1-6
 - définition, 1-6, 3-2
 - fonction et utilisation, 1-6
 - terminaux
 - affectation de terminaux aux commandes et indicateurs, 3-7 à 3-8
 - confirmation des connexions, 3-10
 - connexions obligatoires, recommandées et facultatives, 3-9
 - définition, 3-4 à 3-5

- fenêtre Hiérarchie, 3-15
- modification, 3-6
- sélection, 3-6
- suppression de connexions, 3-10
- types de terminaux
 - connexions nécessaires,
 - recommandées facultatives, 1-8
- connexions nécessaires, 1-8
- connexions obligatoires, 3-9
- connexions recommandées, 1-8, 3-9
- considérations de performances de l’affichage à l’écran, 28-8
- constante boîte de couleurs, 17-6
- constante chaîne de caractères, 17-5
- constante liste déroulante d’erreurs, 17-6
- constante menu déroulant, 17-5
- constante menu déroulant de symboles, 17-6
- constantes, 17-3 à 17-7
 - constante de cluster, 17-4
 - constante de tableau, 17-4
 - constantes numériques, 17-5
 - création automatique, 17-13
 - définies par l’utilisateur, 17-4 à 17-7
 - constante boîte de couleurs, 17-6
 - constante chaîne de caractères, 17-5
 - constante chaîne de caractères (menu local), 17-4
 - constante chemin, 17-7
 - constante liste déroulante d’erreurs, 17-6
 - constante menu déroulant, 17-5
 - constante menu déroulant de symboles, 17-6
 - constantes énumérées, 17-6
 - création, 17-4
 - définir les valeurs, 17-4
 - emplacement dans les palettes, 17-4
 - incrémentatation et
 - décrémentatation, 17-5
 - menu local des constantes numériques (figure), 17-5
 - définition, 17-3
 - menus locaux, 17-5
 - redimensionnement, 17-4, 17-7
 - représentation, 17-7
 - universelles
 - constantes chaîne de caractères, 17-7
 - constantes numériques
 - définition
 - constantes chemin, 17-7
 - constantes définies par l’utilisateur, 17-4 à 17-7
 - constantes énumérées, 17-6
 - constantes numériques, 17-5
 - universelles
 - container ActiveX, 16-2 à 16-5
 - boîte de dialogue Sélectionner un objet ActiveX, 16-2 à 16-4
 - but et utilisation, 16-2
 - commandes ActiveX, 16-2
 - documents ActiveX, 16-2
 - insérer un objet, 16-2
 - contrôle de type d’unités, 9-31 à 9-32
 - contrôle des gammes numériques, 9-7 à 9-9
 - correction des valeurs incorrectes, 9-8
 - option Contraindre, 9-7
 - option Ignorer, 9-7
 - option Interrompre, 9-8
 - convertisseur de panneau de fonction LabWindows/CVI, 25-9 à 25-14
 - copie d’objets, 2-13
 - clonage, 2-13
 - d’autres applications, 2-13
 - entre VIs, 2-13
 - glisser-déposer, 2-9
 - procédure, 2-13
 - Copier, 2-13
 - coude dans un fil, 18-6
 - couleurs
 - préférences, 7-14
 - Couper, 2-13
 - Créer Attribute Node, option, 8-3

<Ctrl-M> raccourci pour passer entre édition et exécution (avertissement), 6-5
 curseurs du graphe, 15-29 à 15-33
 bouton de verrouillage, 15-33
 comportement du type de tableau, 15-30
 composants, 15-29 à 15-30
 contrôle de l'apparence, 15-31 à 15-32
 déplacement, 15-31
 intérêt et utilisation, 15-29
 lecture par programme (exemple), 22-13
 palette curseur (illustrations), 15-29 à 15-30
 pointeurs en croix, 15-31 à 15-32
 styles de point, 15-32
 suppression, 15-30
 Visualiser le nom, 15-32
 curseurs. *Voir* curseurs du graphe

D

Déconnecter ce terminal, option, 3-10
 Déconnecter tous les terminaux, option, 3-10
 décorations
 définition, 8-2
 définitions de type, 24-19 à 24-23
 création, 24-20
 déconnexion, 24-23
 définition, 24-19
 définitions de type cluster, 24-23
 définitions de type strict, 24-20
 enregistrement, 24-20
 exigences des types de données, 24-19
 intérêt et utilisation, 24-21
 recherche, 24-23
 définitions de type strict
 édition (remarque), 24-21
 enregistrement, 24-20
 intérêt et utilisation, 24-20
 dépannage. *Voir* questions concernant LabVIEW

déplacement
 des fils, 18-6 à 18-8
 déplacement d'objets
 clusters, 14-24
 curseurs du graphe, 15-31
 tableaux, 14-14
 désactivation de sections de diagrammes, 4-32
 désassemblage des éléments du cluster, 14-20
 descripteurs de type
 clusters, A-12
 définition, A-7
 format générique, A-7
 généralités, A-7
 tableaux, A-11
 types de données
 enregistrement de quantités
 physiques (remarque), A-11
 exemples, A-10
 non numériques (tableau), A-8
 numériques scalaires (tableau), A-8
 descriptions
 affichage de la fenêtre Aide, 1-7
 création, 2-26 à 2-28
 modification, 2-26
 désélection d'objets, 2-8
 diagramme
 définition, 1-4
 fonction, 1-4
 illustration, 1-5
 placement et dimensionnement de structures sur, 19-5
 préférences des couleurs, 7-14, 7-15
 vue d'ensemble, 1-5
 diagrammes
 composants, 17-1
 constantes, 17-1, 17-3 à 17-7
 fonctions, 17-8 à 17-10
 impression (option Diagramme), 5-7
 option Cadres masqués, 5-7
 option Répéter les étapes de niveau supérieur, 5-7

- insertion des objets, 17-12
- noeuds, 17-1, 17-8 à 17-11
- remplacement des objets, 17-12
- structures, 17-8, 17-11
- terminaux, 17-1 à 17-7
- terminaux de commande et d'indicateur, 17-1, 17-2 à 17-3
- diagrammes parallèles, 28-9 à 28-10
- diagrammes. *Voir aussi* câblage des diagrammes, 17-1 à 17-13
- différences entre les caractères séparateurs, lors du port de VIs d'une plate-forme à une autre, 29-2
- dimensionnement
 - clusters, 14-24
- dimensionnement automatique
 - étiquettes, 2-23
- disque
 - préférences
 - spécification, 7-6
- distribution des VIs
 - considérations, 27-2 à 27-7
 - utiliser Enregistrer avec options, 27-6
- Distribution, menu, 2-12
- documentation
 - conventions utilisées dans le manuel, xxix à xxx
 - organisation du manuel, xxv à xxix
- documentation des VIs, 2-27
- documentation, impression. *Voir* option Imprimer la documentation
- Doigt, outil
 - illustration, 2-5
- données aplaties
 - chaînes de caractères, A-14
 - chemins, A-14
 - clusters, A-15
 - généralités, A-12
 - handles, A-14
 - scalaires, A-13
 - tableaux, A-14

- données big endian, A-13
- données non définies, 4-16

E

- échelles numériques, 9-18 à 9-19
 - étiquettes min et max, 9-18
- Editer l'icône, option, 3-2
- Editer une bibliothèque de VIs
 - boîte de dialogue
 - suppression de fichiers, 2-32
- Editeur d'icône, 3-2
- Editeur de commandes
 - ouverture par double-clic, 7-10
- éditeur de commandes
 - fenêtre des composantes, 24-8 à 24-9
 - illustration, 24-2
 - intérêt et utilisation, 24-2 à 24-3
- Edition
 - Nom utilisateur, option, 7-21
 - option Coller, 2-13
 - option Copier, 2-13
 - option Couper, 2-13
 - option Effacer, 2-14
 - option Ordonnancement de la face-avant, 8-8
- édition des VIs
 - pendant l'exécution des VIs, 4-2 à 4-3
 - options du menu local d'un objet, 4-3
 - options du menu local d'une commande, 4-2
- édition, mode
 - problèmes causés par le masquage des barres de menus et d'outils, 6-5
- effacement. *Voir aussi* suppression enregistrements de données, 4-6
- Effacer, 2-14
- enregistrement définitions de type, 24-20
- enregistrement des commandes
 - personnalisées, 24-5

- enregistrement des données de la face-avant, 4-4 à 4-6
 - afficher les enregistrements, 4-6
 - changer la liaison du fichier d'enregistrement, 4-5
 - effacer les enregistrements marqués, 4-6
 - graphes déroulants (remarque), 4-5
 - marquer des enregistrements pour les effacer, 4-6
 - menu Enregistrement des données, 4-5
 - récupération de données contrôlée par programme, 4-7 à 4-8
 - à l'aide de fonctions d'E/S sur fichier, 4-8
 - accès aux bases de données, 4-7 à 4-8
 - terminaux du halo pour accéder aux données, 4-7 à 4-8
- enregistrement des données en forme d'arbre, A-12
- enregistrement des données. *Voir*
 - enregistrement des données de la face-avant
- enregistrement des VIs, 2-28 à 2-32
 - bibliothèques de VIs, 2-30 à 2-32
 - modification des VIs après enregistrement (avertissement), 2-29
 - option Revenir, 2-29
 - options, 2-28
- Enregistrer, 2-28
- Enregistrer avec options, option, 2-28
- enregistrer des VIs
 - enregistrer pour une distribution, 27-6 à 27-7
- Enregistrer sous, option, 2-28
- Enregistrer une copie sous, option, 2-28
- en-têtes de sections, impression, 5-5
- en-têtes, impression
 - option Imprimer les en-têtes des sections, 5-5
 - option Imprimer un en-tête, 5-9
- entiers, affichage dans d'autres bases, 9-4
- entrée, 1-5
- Entrée, touche
 - arrêt de la saisie de texte, 7-10
- erreurs
 - détectées dans la Boîte de calcul (tableau), 20-9
 - pour VIs invalides
 - erreurs possibles (tableau), 4-11 à 4-13
 - fenêtre Liste des erreurs, 4-10
- erreurs dItd
 - pour VIs invalides
 - fenêtre Liste des erreurs, 4-10
- espace, ajout à la face-avant ou au diagramme, 2-24
- Etiquette
 - option Visualiser, 2-17
 - option, sous-menu Visualiser, 2-3
- Etiquette, option Visualiser, 2-17
- étiquettes, 2-14
 - affichage
 - des étiquettes de fonctions
 - affichage des étiquettes masquées, 2-17
 - asservies, 2-14
 - commandes et indicateurs booléens, 10-3 à 10-4
 - copie de texte, 2-17
 - création, 2-16 à 2-18
 - de fonction (remarque), 2-18
 - définition, 2-14
 - différences lors du port de VIs d'une plate-forme à une autre, 29-2
 - dimensionnement automatique, 2-23
 - libres, 2-14, 2-16
 - modification des attributs de texte, 2-18 à 2-22
 - redimensionnement, 2-23
- étiquettes de fonction, 2-18
- étiquettes min et max, échelles numériques, 9-18
- Exécuter lors de l'ouverture, option, 6-3

- exécution
 - bulles de données pendant l'exécution, sélection, 7-14
 - exécution asservie aux données, 1-6
 - exécution de flux de données, 1-6
 - exécution de la programmation séquentielle dans les structures Séquence, 19-18
 - exécution des VIs en mode Pas à pas, 4-19 à 4-21
 - bouton Exécution détaillée, 4-20 à 4-21
 - bouton Exécution non détaillée, 4-20 à 4-21
 - bouton Pause, 4-19
 - bouton Sortie, 4-20 à 4-21
 - exécution des VIs, 4-19
 - exécution en mode Animation, 4-22 à 4-24
 - exemple, 4-20
 - lecture des chaînes d'appel, 4-21
 - utilisation des boutons Pas à pas, 4-21
 - exécution des VIs. *Voir aussi* performances
 - arrêt des VIs, 4-4
 - enregistrement des données de la face-avant, 4-4 à 4-6
 - exécution des VIs, 4-1 à 4-3
 - boutons pour l'exécution, 4-1
 - édition pendant l'exécution, 4-2 à 4-3
 - exécution répétée, 4-4
 - plusieurs VIs, 4-2
 - récupération de données contrôlée par programme, 4-7 à 4-8
 - à l'aide de fonctions d'E/S de fichier, 4-8
 - accès aux bases de données, 4-7 à 4-8
 - exécution en mode Animation. *Voir* exécution en mode Animation, 4-22 à 4-24
 - bouton Exécution en mode Animation, 4-22
 - exemple, 4-23
 - glyphes d'exécution, 4-23
 - exécution multithread, 26-2 à 26-7
 - aperçu, 26-2
 - coopératif, 26-2
 - HP-UX et Solaris 1, 26-3
 - Macintosh et Windows 3.1, 26-3
 - Mutex (sémaphore), 26-19
 - préemptif, 26-2
 - système d'exécution de base, 26-3
 - système d'exécution du G, 26-3
 - systèmes d'exécution multiples, 26-4 à 26-7
 - Windows 95/NT, Solaris 2 et PowerMAX, 26-3
 - exécution multithread. *Voir aussi* exécution à un seul thread, 26-2
 - exécution ré-entrante
 - exemples
 - VI de stockage qui n'est pas sensé partager ses données, 26-15 à 26-16
 - VIs en attente, 26-14 à 26-15
 - Exécution ré-entrante, option, 6-3
 - exécution, mode
 - problèmes causés par le masquage des barres de menus et d'outils (Avertissement), 6-5
 - exécution, palette. *Voir* barre d'outils, exécution
- ## F
- face-avant
 - application des modifications des commandes personnalisées, 24-4
 - Associer une touche, boîte de dialogue, 8-6
 - commandes de boîte de dialogue, 8-9 à 8-10
 - construction, 8-1 à 8-11
 - définition, 1-3
 - enregistrement des données, 4-4 à 4-6

- illustration, 1-3
- importation de graphiques, 8-10
- impression (option Face-avant), 5-7
- options de commandes et indicateurs, 8-2 à 8-6
- ordre des panneaux, 8-8 à 8-9
- préférences des couleurs, 7-15
- présentation, 1-3
- règles d'utilisation de la mémoire, 28-20 à 28-22
- remplacement des commandes, 8-4 à 8-6
- fenêtre d'aide
 - aide du câblage, 18-5
 - Attribute Nodes, 22-13
- fenêtre de profil, 28-1 à 28-6
- fenêtre Hiérarchie
 - visualisation pendant une interruption d'exécution, 4-31
- Fermer après l'exécution si fermé à l'origine, option, 6-2
- Fichier
 - Importer une image, option, 8-10
 - option Editer une bibliothèque de VIs, 2-32
 - options d'enregistrement, 2-28
- Fichier, menu
 - option Revenir, 2-29
- fichiers HTML (Hypertext Markup Language), impression de descriptions de commandes et de VIs vers, 5-10 à 5-15
- fichiers JPEG, enregistrer des images sous forme de, 5-11 à 5-13
- fichiers PNG (Portable Network Graphics), 5-11
- fichiers RTF (Rich Text Format), impression de descriptions de commandes et de VIs vers, 5-10 à 5-15
- fichiers, gestion
 - stockage de VIs dans des bibliothèques, 2-30
- fils de liaison
 - branches, 18-6
 - coudes, 18-6
 - jonctions, 18-6
 - portions, 18-6
 - techniques de base, 1-5
- Flèche, outil
 - déplacement d'objets, 2-9 à 2-10
 - illustration, 2-5
 - sélection d'objets, 2-9 à 2-10
- flux de données
 - comparaison avec la programmation séquentielle, 1-5
 - définition, 1-5
- fonctionnement multitâche, 26-1 à 26-16
 - accorder des priorités aux tâches du même niveau de priorité, 26-7
 - commutation des tâches, 26-3
 - coopératif, 26-1 à 26-2
 - exécution ré-entrante
 - exemples, 26-14 à 26-16
 - fonctions Attendre pour accorder des priorités aux tâches, 26-8
 - noeuds synchrones de blocage, 26-7
 - préemptif, 26-2
- fonctions, 17-8 à 17-10
 - Assembler, 14-25
 - Assembler par nom
 - accès aux éléments du cluster, 24-23
 - “Assembler par nom”
 - assemblage clusters, 14-26 à 14-28
 - “Cluster en tableau”, 14-32
 - “Construire un tableau”, 17-10
 - Désassembler par nom
 - accès aux éléments du cluster, 24-23
 - “Fermer une référence de VI”, 21-10
- fonctions extensibles, 17-10
 - “Ouvrir une référence d'application”, 21-4
 - “Ouvrir une référence de VI”, 12-5, 21-4, 21-10

- Séparer, 14-30
 - “Séparer par nom”
 - désassemblage des éléments du cluster, 14-30 à 14-32
 - “Tableau en cluster”, 14-2
- fonctions Attendre
 - exemple d’exécution ré-entrante, 26-14 à 26-15
 - priorité des tâches, 26-8
- fonctions d’E/S sur fichier, pour récupérer des données enregistrées, 4-8
- fonctions d’occurrence, 26-21
- fonctions de gestion de la sélection de menu, 6-15 à 6-20
 - activer le repérage des menus, 6-16
 - élément de menu sélectionné, 6-15
 - fonctions de menu dynamiques, 6-16
 - insérer des éléments de menus, 6-16
- fonctions de synchronisation. *Voir aussi* sémaphores, 26-21
- fonctions extensibles, 17-10
- Fonte, menu, 2-19
- fontes
 - différences lors du port de VIs d’une plate-forme à une autre, 29-3 à 29-5
 - étiquettes, 2-18 à 2-22
 - fontes prédéfinies de LabVIEW, 29-4
 - préférences, 7-16
- format d’enregistrement des données numériques en précision étendue, A-2
- format Personnalisé, boîte de dialogue
 - Imprimer la documentation, 5-6
- formatage du temps absolu et de la date
 - affichages numériques, 9-10 à 9-11
- formats d’enregistrement des données
 - booléens, A-1
 - chaînes de caractères, A-5
 - chemins, A-5
 - clusters, A-5 à A-7
 - handles, A-5
 - nombres, A-3

- numérique
 - double précision, A-3
 - entier long, A-3
 - entier mot, A-3
 - entier octet, A-3
 - simple précision, A-3

- tableaux, A-4

- formats d’enregistrement des données
 - numérique
 - précision étendue, A-2

G

- gamme des données
 - contrôle de la gamme de données des booléens, 10-4
- gestion de fichiers
 - arranger des fichiers dans des bibliothèques de VIs, 27-1 à 27-2
 - sauvegarde de fichiers, 27-2
- glisser-déposer, 2-8 à 2-9
- glissières à valeurs multiples, 9-19
- glissières pleines et à valeurs multiples, 9-19
- graphe d’intensité, 15-39
 - options, 15-39
 - type de données, 15-39
- graphes à tracés multiples
 - Voir* forme d’onde et graphes XY, xxxi
- graphes déroulants, 15-22 à 15-28
 - curseurs non pris en charge, 15-24
 - enregistrement de données de la face-avant (remarque), 4-5
 - options, 15-24 à 15-28
 - mode de mise à jour, 15-25 à 15-27
 - mode de mise à jour de l’oscilloscope, 15-26, 15-27
 - mode de mise à jour du graphe déroulant, 15-26
 - tracés empilés comparés aux tracés superposés, 15-28
- types de données, 15-22 à 15-23

graphes déroulants d'intensité, 15-34 à 15-38
 illustrations, 15-35 à 15-36
 options, 15-36 à 15-38
 définition de la représentation des couleurs, 15-38
 historique de données, 15-37
 menu local, 15-37

graphes XY et en forme d'onde
 options du graphe
 options d'échelle, 15-12 à 15-17

graphes XY et oscilloscopiques
 création de graphe à tracé unique, 15-3 à 15-4
 types de données du graphe oscilloscopique, 15-3
 types de données du graphe XY, 15-4
 création de graphe multicourbes, 15-5
 types de données du graphe XY, 15-11
 définition, 15-2
 illustration, 15-2
 options du graphe, 15-11 à 15-21
 illustration, 15-12
 option Légende, 15-19
 options d'échelle, 15-12

graphiques
 importation, 8-10

graphiques. *Voir aussi* composantes décoratives
 exportation
 format JPEG, 5-11 à 5-14
 format PNG, 5-11 à 5-14

H

handles
 données aplaties, A-14
 formats d'enregistrement des données, A-5

hiérarchie des VIs
 affichage, 3-15
 importance, 3-1

hiérarchie du VI, impression, 5-7

Hiérarchie, fenêtre, 3-15

historique
 préférences, 7-18 à 7-21

historique du développement d'un VI
 fenêtre Historique, 27-9 à 27-13
 impression des informations de l'historique, 27-13
 numéros de révision, 27-11 à 27-12
 options associées des boîtes de dialogue
 Configuration du VI et Préférences, 27-13
 réinitialisation des informations de l'historique, 27-12

historique du VI. *Voir aussi* boîte de dialogue Préférences de l'historique
 impression, 5-7

I

icône de fonction Indexation du tableau, 14-19

icône du format Documentation complète, 5-4

icônes
 création, 3-2 à 3-4
 outils, 3-3 à 3-4
 pour des commandes, 24-5
 création d'icônes couleur, 3-3 à 3-4
 définition, 1-6, 3-2
 fonction, 1-6
 icône par défaut, 1-6
 impression (option Icône et description), 5-6

icônes couleur, création, 3-3

images
 différences lors du port de VIs d'une plate-forme à une autre, 29-2

importation de graphiques, 8-10

Importer une image, commande, 8-10

- impression
 - préférences, 7-17 à 7-18
- impression (option Commandes), 5-7
 - option Descriptions, 5-7
 - option Inclure des informations sur le type de données, 5-7
- impression contrôlée par programme, 5-7
 - affinage des impressions, 5-9
 - contrôle du moment auquel ont lieu les impressions, 5-8
 - définition de la mise en page, 5-9 à 5-10
- impression dans LabVIEW
 - questions concernant, B-8 à B-10
- impression des VIs, 5-1 à 5-16
 - autres méthodes d'impression, 5-10
 - choix des options d'impression, 5-2 à 5-7
 - boîte de dialogue Imprimer la documentation, 5-4
 - choix d'options de mise en page, 5-5
 - création de paramètres d'impression personnalisée, 5-6 à 5-7
 - définition des formats d'impression, 5-4
 - impression des en-têtes de section, 5-5
 - configuration de l'impression, 5-2
 - fenêtre active (option Imprimer la fenêtre), 5-2
 - impression contrôlée par programme, 5-7 à 5-10
 - affinage des impressions, 5-9
 - contrôle du moment auquel ont lieu les impressions, 5-8
 - définition de la mise en page, 5-9 à 5-10
 - impression en PostScript, 5-2
 - présentation, 5-1
 - VI "AESend Print Document", 5-1
 - VI "Exec. système", 5-1
 - VIs Port série, 5-1
- impression en PostScript, 5-2
- impression/exportation vers des fichiers RTF ou HTML, 5-10 à 5-15
 - boîte de dialogue Imprimer la documentation, 5-11
 - pour fichier HTML (figure), 5-13
 - pour fichier RTF (figure), 5-12
 - exemples de noms de fichiers d'image (tableau), 5-15
 - format GIF, 5-14
 - format JPEG, 5-11 à 5-14
 - noms de fichiers résultants (tableau), 5-14
 - format PNG, 5-13
- Imprimer les en-têtes des sections, 5-5
- indicateur d'erreur de dépassement de gamme, 4-17
- indicateurs
 - affectation de terminaux, 3-7 à 3-8
 - changer en commandes, 8-3, 18-11
 - couper, copier et coller, 8-3
 - définition, 1-3
 - masquer, 2-14
 - options communes, 8-2 à 8-4
 - remplacement, 8-4 à 8-6
 - suppression interdite, 2-14
- indicateurs de graphe
 - création de graphe multicourbes, 15-5
 - curseurs du graphe, 15-29 à 15-33
 - définition, 15-1
 - graphe d'intensité, 15-39
 - options, 15-39
 - type de données, 15-39
 - graphes XY et oscilloscopiques, 15-2
 - illustration, 15-2
 - menu local d'échelle, 9-14
 - options du graphe, 15-11
 - illustration, 15-12
 - menu local du graphe, 15-12

- options
 - Légende, 15-19
 - options d'échelle, 15-12 à 15-17
 - options de défilement et de zoom, 15-17 à 15-19
- questions concernant, B-1 à B-4
- indicateurs de graphe déroulant
 - attributs
 - attribut Couleur du tracé, 22-9 à 22-10
 - attribut Courbe active, 22-10
 - graphe déroulant
 - modes de mise à jour, 15-25 à 15-27
 - options, 15-24 à 15-28
 - tracés empilés contre tracés superposés, 15-27
 - types de données, 15-22 à 15-23
 - graphe déroulant d'intensité, 15-34 à 15-38
 - définition de la représentation des couleurs, 15-38
 - illustrations, 15-35 à 15-36
 - menu local, 15-37
 - options, 15-36
 - graphe oscilloscopique, 15-22
 - questions concernant, B-1 à B-4
- indicateurs et commandes de graphe et de graphe déroulant, 15-1 à 15-39
- indicateurs. *Voir* commandes et indicateurs
 - erreurs de dépassement de gamme (remarque), 4-17
- indicateurs. *Voir* commandes et indicateurs
 - propagation par opérations à virgule flottante, 4-16
- indices des tableaux, 14-2 à 14-5
- Info de l'échelle des Z, 15-38
- info-labview (groupe d'utilisateurs), B-11
- informations d'aide
 - création de vos propres fichiers d'aide, 5-15 à 5-16
- Interrompre à l'appel, option, 6-3

- interruption de l'exécution, 4-29 à 4-31
 - bouton Exécution, 4-31
 - bouton Retour au début, 4-31
 - bouton Revenir au VI appelant, 4-31
 - options, 4-29
 - pendant la mise au point, 4-29
 - reconnaître une interruption automatique, 4-30
 - utilisation des boutons de la barre d'outils pendant l'interruption, 4-31
 - visualisation de la fenêtre Hiérarchie pendant l'interruption, 4-31
- interruptions
 - gestion, B-14

J

- jonction (de fils), 18-6

L

- Libérer la mémoire dès que possible, option, 7-7
- ligne de division, pour les commandes des listes déroulantes, 13-7
- localisation des chaînes de caractères, 5-15

M

- Main, outil, 2-5
- mémoire
 - questions concernant, B-5
 - spécification des préférences, 7-7
- menu déroulant Chaîne d'appel, 4-21
- menu Edition
 - option Editer la commande, 24-2, 24-16
 - option Supprimer les fils incorrects, 18-6, 18-8, 18-11

- menu Enregistrement des données, 4-5 à 4-6
 - illustration, 4-5
 - opération d'enregistrement des données, 4-4
 - option Changer la liaison du fichier d'enregistrement, 4-5
 - option Purger les données, 4-6
- menu Exécuter
 - Passer à l'option mode Personnalisé, 24-6
 - Passer en mode Edition, 24-6
- menu Exécution
 - Abandonner, 4-4
 - Effacer la liaison du fichier d'enregistrement, 4-6
 - Enregistrer à la fin de l'exécution, 4-5
 - Exécuter, 4-1
 - Imprimer à la fin de l'exécution, 5-8
 - Interrompre à l'appel, 4-29
 - Récupérer, 4-6
 - sous-menu Enregistrement des données, 4-5 à 4-6
- menu Fenêtres
 - Visualiser la fenêtre des composantes, 24-8
- menu Fichier
 - Configuration de l'imprimante, 5-2
 - Mise en page, 5-2
 - option Appliquer les modifications, 24-4
 - option Enregistrement, 24-5
 - option Enregistrer, 24-20
 - option Revenir, 24-12
- menu local d'échelle, 9-14 à 9-15
 - Espacement des repères, 9-14
 - Format & Précision, 9-14
 - illustration, 9-14
 - Représentation, 9-15
 - Style, 9-15
- menu local de chaîne de caractères, 11-3
- menu local de l'échantillon tracé, 15-20
- menu local de la zone d'affichage de texte, 9-19
- menu local du graphe, 15-12
- Menu local, outil, 2-5
- menu Opérations sur les données
 - options du menu local d'un objet, 4-3
 - options du menu local d'une commande, 4-2 à 4-3
- menu Projet
 - Importer des commandes ActiveX, 16-5
 - Sous-VIs appelés par ce VI, 4-15
 - Sous-VIs non ouverts, 4-15
- menus
 - barre de menus, 2-6
 - déroulants, 2-6
 - menus locaux, 2-6
- menus de LabVIEW, 2-1 à 2-3
 - menus locaux, 2-1 à 2-3
- menus locaux
 - accès, 1-4
 - définition, 2-6
 - utilisation, 2-6
- <meta-M> raccourci pour passer entre édition et exécution (avertissement), 6-5
- mise au point des VIs, 4-9 à 4-32
 - désactivation de sections de diagrammes, 4-32
 - désactivation des fonctions de mise au point, 4-31
 - exécution des VIs en mode Pas à pas, 4-19 à 4-21
 - bouton Exécution détaillée, 4-20 à 4-21
 - bouton Exécution non détaillée, 4-20 à 4-21
 - bouton Pause, 4-19
 - bouton Sortie, 4-20 à 4-21
 - exécution des VIs, 4-19
 - exemple, 4-20
 - utilisation des boutons Pas à pas, 4-21
 - exécution en mode Animation, 4-22 à 4-24

interruption de l'exécution, 4-29 à 4-31
 bouton Exécution, 4-31
 bouton Retour au début, 4-31
 bouton Revenir au VI appelant, 4-31
 options, 4-30
 reconnaitre une interruption
 automatique, 4-30
 utilisation des boutons de la barre
 d'outils pendant
 l'interruption, 4-31
 visualisation de la fenêtre Hiérarchie
 pendant l'interruption, 4-31
 lecture des chaînes d'appel, 4-21
 mise en place de points d'arrêt,
 4-27 à 4-29
 affichage des points d'arrêt
 (tableau), 4-28
 exemple, 4-29
 outil Sonde
 création de sonde, 4-26 à 4-27
 utilisation, 4-24 à 4-25
 VIs exécutables, 4-14 à 4-17
 comprendre les mises en garde, 4-16
 mesures de résolution de problème,
 4-14 à 4-16
 reconnaitre des données non
 définies, 4-16
 VIs invalides
 correction des erreurs de
 dépassement de gamme des VIs
 invalides, 4-14
 localisation des erreurs, 4-10
 messages d'erreur (tableau),
 4-11 à 4-13
 raisons courantes pour les VIs
 invalides, 4-10
 réparation des VIs invalides,
 4-9 à 4-13
 mise en page, contrôlée par programme,
 5-9 à 5-10
 mode de mise à jour de l'oscilloscope, 15-26

mode de mise à jour du graphe
 déroulant, 15-26
 mode Personnalisé, 24-6 à 24-19
 commandes en composantes,
 24-16 à 24-17
 composantes de texte, 24-15
 composantes indépendantes, 24-7 à 24-8
 fenêtre des composantes de l'éditeur de
 commandes, 24-8 à 24-9
 menus locaux des différentes
 composantes, 24-10
 sélection, 24-6
 multithreading coopératif, 26-2

N

NaN (pas un nombre)
 erreurs de dépassement de gamme
 (remarque), 4-17
 propagation par les opérations à virgule
 flottante, 4-16
 Niveau de coopération, option, 7-9
 niveau de priorité de type
 sous-programme, 26-11
 noeud
 définition, 1-5
 flux de données, 1-5
 noeud "Appeler une fonction", 25-3 à 25-8
 appel de fonctions qui attendent d'autres
 types de données, 25-8
 boîte de dialogue "Appeler une
 fonction", 25-4
 création d'une liste de paramètres, 25-7
 type de données chaîne de
 caractères, 25-7
 type de données numérique, 25-6
 type de données tableau, 25-6
 type de données vide, 25-6
 présentation, 25-2
 noeud de méthode, 21-4 à 21-8
 noeud de propriété, 21-5 à 21-6

noeuds. *Voir aussi* structures
 définition, 17-1
 noeuds synchrones de blocage, 26-7
 utilisation des noeuds de propriété et de
 méthode avec les références
 d'application et de VI, 21-4 à 21-7
 vue d'ensemble, 17-8

Nom utilisateur, option, 7-21

nombres 32 bits simple précision. *Voir*
 nombres à virgule flottante en simple
 précision

nombres à virgule flottante en précision
 étendue
 format de précision étendue de
 LabVIEW, A-14

Numérique, palette, 8-3

numéros de révision pour les VIs,
 27-11 à 27-12

O

objets

alignement, 2-11

copie, 2-13

déplacement
 option Placer derrière, 2-11
 option Placer devant, 2-10
 option Placer en arrière-plan, 2-11
 option Placer en premier plan, 2-11
 outil Flèche, 2-9

étiquettes, 2-14
 création, 2-16 à 2-18
 modification des attributs de texte,
 2-18 à 2-22

menus, 2-1

menus locaux, 2-6

méthodes de déplacement, 2-9 à 2-10

option Placer derrière, 2-11

option Placer devant, 2-10

option Placer en arrière-plan, 2-11

option Placer en premier plan, 2-11

outil Flèche, 2-9

redimensionnement, 2-22 à 2-24

répartition, 2-12

sélection, 2-7 à 2-8

suppression, 2-14

opérations à virgule flottante et données non
 définies, 4-16

Opérations sur les données, sous-menu
 mode édition, 8-3
 option Description, 2-26

option Légende pour les indicateurs de
 graphe, 15-19

option Préférences, 7-1

options
 <Retour>, 6-4
 Activer l'indexation, 19-8
 Adapter à la source, 17-7
 Affichage de l'indice, 14-11
 Affichage des codes '\'
Voir aussi codes à barre oblique
 inverse (\), 11-3
 Affichage des codes '\'. *Voir aussi* codes
 à barre oblique inverse (\), 11-2
 erreurs de dépassement de gamme
 (remarque), 4-17
 propagation par opérations à virgule
 flottante, 4-16

Affichage hexadécimal, 11-6

Affichage style mot de passe, 11-6

Ajouter un élément, 19-11

Ajouter un élément après
 commandes de type énuméré, 13-14
 menu déroulant, 13-14
 menu local de la zone d'affichage de
 texte, 9-19

Ajouter un élément avant
 commandes de type énuméré, 13-14
 menu déroulant, 13-14
 menu local de la zone d'affichage de
 texte, 9-19

Ajouter un paramètre après, 25-5

- Ajouter un registre à décalage, 19-9
- Ajouter un repère, 9-17 à 9-18
- Ajouter une aiguille, 9-21
- Ajouter une entrée, 17-10, 20-2
- Ajouter une entrée, menu local de câblage, 17-10
- Ajouter une glissière, 9-20
- Ajouter une sortie, 20-2
- Ajouter une variable locale de séquence, 19-19
- Ajuster à la taille du texte, 17-4
- Ajuster l'échelle, 15-17
- Ajuster la taille de la face-avant, 5-5
- AppleEvent, menu Communications, 5-1
- Appliquer les modifications, 24-4
- Attaché au point, 15-33
- Atteindre le curseur, 15-32
- Autoriser l'accès à la base de données, 4-7
- Autoriser le glissement, 15-33
- Barre de défilement, 11-3
- Base du remplissage, 15-21
- Boîte de dialogue, 6-4
- Centrage automatique, 6-5
- Centrer, 15-32
- Changer en commande, 14-22, 18-11
- Changer en écriture, 21-6, 22-2
- Changer en entrée, 20-3
- Changer en indicateur, 14-22, 18-11
- Changer en lecture, 21-6, 22-2 à 22-3
- Changer en sortie, 20-3
- Changer la liaison du fichier d'enregistrement, 4-5 à 4-6
- Changer la variable globale en écriture, 23-3
- Changer la variable globale en lecture, 23-3
- Changer la variable locale en écriture, 23-4
- Changer la variable locale en lecture, 23-4
- Chemin valide, 12-2
- Coller des données, 11-9
- Commandes, boîte de dialogue Paramètres d'impression personnalisée, 5-7
- Commencer la sélection, 14-16, 14-17, 15-30
- Communication, menu Fonctions, 5-1
- Configuration de l'imprimante, 5-2
- Configuration du nœud du sous-VI, 4-30
- Copier dans le presse-papiers, 24-11
 - composantes décoratives menu local, 24-10
- Copier des données, 11-9
 - outil Sonde, 4-25
- Couleur, 15-21
- Couleur du tracé, 22-10
- Couper des données, 11-9, 14-13
- Créer Attribute Node, 22-1, 22-4
- Créer un indicateur, 17-13
- Créer un objet à partir d'un fichier, 16-3, 16-4
- Créer une commande, 17-13
 - boîte de dialogue Sélectionner un objet ActiveX, 16-3 à 16-4
 - menu local sur un terminal, 17-13
- Créer une constante, 17-4, 17-13
- Déconnecter de la déf. de type, 24-23
- Désactiver cet élément
 - commandes des listes déroulantes, 13-7
 - commandes des menus déroulants, 13-7
- Désactiver l'indexation, 19-8
- Diagramme, boîte de dialogue Paramètres d'impression personnalisée, 5-7
- Distribution pour application, 27-7
- Distribution pour développement, 27-7
- Dupliquer cette condition, 19-21
- Editer la commande, 24-2, 24-16
- Effacer la liaison du fichier d'enregistrement, 4-6
- Élément image, 24-13

- Empiler les tracés, 15-28
- Encadrer la face-avant d'une bordure, 5-10
- Enregistrement, 24-5
- Enregistrement personnalisé, 27-7
- Enregistrer, 4-5, 24-20
- Enregistrer à la fin de l'exécution, 4-5
- Enregistrer avec options, 2-28, 27-3, 27-6
- Enregistrer sous, 2-28
- Enregistrer une copie sous, 2-28
- Epaisseur de ligne, 15-20
- Espacement des repères, 15-13
 - menu local de la glissière, 9-17
- Etiquettes texte, 9-18
- Exécuter lors de l'ouverture, 6-3
- Face-avant, boîte de dialogue Paramètres d'impression personnalisée, 5-7
- Facteurs d'échelle, 15-15
- Fermer après l'exécution si fermé à l'origine, 6-2
- Fonte de dialogue, 2-18
- Format & Précision
 - affichages numériques, 9-9 à 9-11
 - boîte de dialogue Format & Précision, 9-9
 - exemple, 9-10
 - temps absolu et date, 9-10 à 9-11
 - Attribute Nodes, 22-5
- Format et précision
 - des menus locaux du graphe, 15-15
- format Icône, description, face-avant et diagramme, 5-4
- format Utilisation comme sous-VI, 5-4
- format Utilisation de la face-avant, 5-4
- Formatage, indicateurs de graphe, 15-14
- Gamme des données, 4-17, 9-15
- Hiérarchie du VI, 5-7
- Historique du VI, 5-7
- Icône et description, 5-6
 - Connecteur et icône du VI, 5-6
 - Description du VI, 5-6
- Imprimer à la fin de l'exécution, 5-8
- Imprimer la documentation, 5-1, 5-3
- Imprimer la face-avant à la fin de l'exécution du VI, 5-9
- Imprimer la fenêtre, 5-1, 5-2
- Imprimer les en-têtes des sections, boîte de dialogue Imprimer la documentation, 5-5
- Imprimer un en-tête, 5-5, 5-9
- Insérer, 17-12
- Insérer un objet ActiveX, 16-2
- Interpolation, 15-21
- Interpoler les couleurs, 9-24
- Interrompre à l'appel
 - menu Exécution, 4-29
- interrompre à l'appel, 6-3
- Libre, 15-33
- Limiter à une seule ligne, 11-6
- Liste des erreurs, 18-8
- Liste des sous-VIs, boîte de dialogue Paramètres d'impression personnalisée, 5-7
- Longueur de l'historique du graphe déroulant, 15-37
- Marges de page, 5-10
- Mettre à jour à partir de la déf. de type, 24-22
- Mise à jour automatique de la déf. de type, 24-22
- Mise à l'échelle automatique des X, 15-18
- Mise à l'échelle automatique des Y, 15-18
- Mise en page, 5-2
- Mode de mise à jour, 15-25
- option d'exécution
 - priorité, 26-9
 - système d'exécution préféré, 26-8
- Options de remplissage, 9-19
- Ordonnancement de la face-avant, B-15
- Ordonner le cluster, 4-15, 14-23
- Pas un chemin, 12-2
- Passer en mode Edition, 24-6

- Purger les données, 4-6
- Rafraîchissement progressif, 15-13
- Rechercher le fil de liaison, 4-25
- Récupérer, 4-6
- Réinitialiser avec les paramètres par défaut, 14-12, 14-13
 - outil Sonde, 4-25
- Réinitialiser les valeurs par défaut, 14-22
- Remplacer, 17-12
- Remplir jusqu'à la valeur ci-dessous, 9-20
- Remplir jusqu'à la valeur ci-dessus, 9-20
- Représentation, 9-5, 9-15, 15-15
- Revenir, 2-29, 24-12
- Sauts de page entre sections, 5-5
- Sonde personnalisée, 4-26
- Sous-VIs appelés par ce VI, 4-15
- Style
 - menu local d'échelle, 9-15
- Style d'échelle, 15-14
- Style de ligne, 15-20
- Style de point, 15-20
- Supprimer ce paramètre, 25-5
- Supprimer cette condition, 19-22
- Supprimer les fils incorrects, 4-9, 18-8, 18-11
- Supprimer tout, 19-11
- Supprimer un élément, 19-11
- Supprimer une dimension, 14-8
- Supprimer une entrée, 17-10
- système d'exécution préféré, 26-5
- Taille du cluster, 14-29
- Taille initiale, 24-12
- Tailles indépendantes, 24-15
- Terminer la sélection, 14-16 à 14-17, 15-30
- Texte booléen, 10-4
- Tous les éléments, 22-5
- Tracés à barres, 15-21
- Tracés communs, 15-20
- Transposer le tableau
 - graphe d'intensité, 15-39
 - menu local du graphe, 15-6, 15-12
 - menu local du graphe déroulant, 15-23
- Trouver la commande, 22-6
- Trouver le terminal, 22-6
- Verrouiller au tracé, 15-33
- Verrouiller l'aide, 1-7
- Verrouiller le texte au centre, 10-3
- Virtual Instrument, 21-4, 21-8
- VIs modifiés, 27-6
- Visualiser affichage numérique
 - commandes et indicateurs à glissière, 9-14
- Visualiser Etiquette, 2-3
- Visualiser la condition en, 19-20
- Visualiser la face-avant lors de l'appel, 6-2
- Visualiser la face-avant lors du chargement, 6-2
- Visualiser la fenêtre des composantes, 24-8
- Visualiser la sélection, 14-15
- Visualiser le bouton Abandonner, 4-4
- Visualiser le connecteur, 1-6
- Visualiser le dernier élément, 14-14
- Visualiser le nom, 15-32
- Visualiser les mises en garde, 4-10
- Visualiser les terminaux, 17-1
- Visualiser terminaux, 17-9
- options d'échelle, indicateurs de graphe, 15-12 à 15-17
- Options d'exécution, 6-2 à 6-3
 - option Exécuter lors de l'ouverture, 6-3
 - option Exécution ré-entrante, 6-3
 - option Fermer après l'exécution si fermé à l'origine, 6-2
 - option Interrompre à l'appel, 6-3
 - option Priorité, 6-3

- Visualiser la face-avant lors de l'appel, option, 6-2
- Visualiser la face-avant lors du chargement, option, 6-2
- options de défilement, indicateurs de graphe, 15-17 à 15-19
- options de gamme des commandes et indicateurs numériques, 9-6
 - boîte de dialogue Gamme des données, 9-6
 - contrôle des gammes numériques, 9-7 à 9-9
 - correction des valeurs incorrectes, 9-8
 - option Contraindre, 9-7
 - option Ignorer, 9-7
 - option Interrompre, 9-8
 - gamme des nombres à virgule flottante en précision étendue (remarque), 9-6
 - nombres à virgule flottante (tableau), 9-6
- Options de grille, 15-15
- options de zoom, indicateurs de graphe, 15-17 à 15-19
- Options des fenêtres
 - case Autoriser les menus locaux pendant l'exécution, 6-4
 - option Activer l'enregistrement/l'impression après l'exécution, 6-5
 - option Ajuster à la taille de l'écran, 6-5
 - option Boîte de dialogue, 6-4
 - option Centrage automatique, 6-5
 - option Visualiser la barre d'outils, 6-5
- options du menu local d'une valeur numérique, 9-4
- Ordonnancement de la face-avant, option, 8-8
- outil Bobine
 - amorçe, 18-1
 - désactivé dans l'éditeur de commandes, 24-8
- outil Doigt, 9-2
- outil Point d'arrêt, 4-27

- outil Sonde
 - création de sonde, 4-26 à 4-27
 - utilisation, 4-24 à 4-27

- outils
 - changer d'outil, 2-5
 - illustration, 2-4
 - palette, 2-4 à 2-5

P

- palette Action mécanique, 10-4
- palette Boîte de calcul, 20-1
- palette Booléen, 10-3
- palette Chemin & Refnum, 12-1, 21-4
- palette Commandes
 - accès au squelette du cluster, 14-21
 - ajout de commandes personnalisées, 24-1 Booléen, 10-1
 - option Sélectionner une commande, 24-5
- palette Graphe, xxvii, II-2, 15-1 à 15-33
- palette Tableaux et Clusters, xxvii, II-1, 14-1
 - sélection de commandes et indicateurs, 1-3
- palette Décorations, 5-9, 24-17
- palette Fonctions
 - constantes disponibles dans, 17-4
 - illustration
 - option Communication, 5-1
- palette Graphe, xxvii, II-2, 15-1 à 15-33
- palette Numérique
 - illustration, 9-1
- palette Sémaphores, 26-21
- palette Sonde personnalisée, 4-26
- palette Tableaux et clusters, 14-5
- palette Variable globale, 23-1
- palette Variable locale, 23-4
- panneaux locaux
 - création, 6-1 à 6-2
- Passer à l'option mode Personnalisé, 24-6
- Pattern, option, 3-6

- performance
 - préférences
 - spécification, 7-6
- Performance Profiler, 28-1 à 28-6
- performances
 - accélération des VIs, 28-6 à 28-13
 - affichage à l'écran, 28-8
 - calculs superflus dans les boucles, 28-11 à 28-13
 - diagrammes parallèles, 28-9 à 28-10
 - entrée/sortie, 28-7 à 28-8
 - temps machine d'accès aux sous-VIs, 28-11
 - profil de performances des VIs, 28-1 à 28-6
 - structures de données efficaces, 28-32 à 28-40
 - approche évidente, 28-36
 - approches alternatives, 28-37 à 28-39
 - éviter les types de données compliqués, 28-34 à 28-35
 - table de chaînes de caractères globale statique, 28-39
 - table globale de types de données mélangés, 28-35 à 28-39
 - utilisation de la mémoire, 28-13 à 28-31
 - composantes des VIs, 28-15
 - concepts de base, 28-14 à 28-18
 - contrôle, 28-18 à 28-19
 - faces-avant, 28-20 à 28-22
 - généralités, 28-13
 - libération par LabVIEW, 28-23 à 28-24
 - mémoire virtuelle, 28-14
 - programmation par flux de données et buffers de données, 28-16 à 28-18
 - règles générales pour une amélioration, 28-19 à 28-20
 - réutilisation du buffer d'entrée par la sortie, 28-24
 - réutilisation par les sous-VIs de la mémoire des données, 28-22
 - variables globales, 28-23
 - variables locales, 28-22
- Pinceau, outil, 2-5, 2-24
- Pipette, outil, 2-5
- Point d'arrêt, outil, 2-5
- pointeur Hige (remarque), 25-7
- points, 14-20, 15-4
- points d'arrêt, mise en place, 4-27 à 4-29
 - affichage des points d'arrêt (tableau), 4-28
 - exemple, 4-29
- points de coercition, 7-15
- portabilité, 29-1 à 29-3
 - différences de résolution et de fonte, 29-3 à 29-5
 - étiquettes, 29-4
 - fontes prédéfinies, 29-4
 - différences entre les caractères séparateurs, 29-2
 - différences entre les images, 29-5
 - facilité de port d'une plate-forme à une autre, 29-2
 - généralités, 29-1
 - VIs non portables, 29-1 à 29-2
- ports
 - entrée, 1-5
 - sortie, 1-5
- préférences
 - boîtes de dialogue, 7-1 à 7-24
 - chemin, 7-2 à 7-6
 - boîte de dialogue, 7-2
 - Chemin du recherche du VI, 7-4 à 7-6
 - répertoire de bibliothèque, 7-3
 - répertoire par défaut, 7-4
 - répertoire temporaire, 7-4
 - couleurs, 7-14

- diagramme, 7-12
 - diverses, 7-23
 - face-avant, 7-10 à 7-11
 - ouverture de commande par double-clic, 7-10
 - vitesse de clignotement, 7-11
 - fontes, 7-16 à 7-17
 - heure et date, 7-22
 - historique, 7-18 à 7-21
 - impression, 7-17 à 7-18
 - mise au point, 7-13
 - performance et disk
 - option Libérer la mémoire dès que possible, 7-7
 - performance et disque, 7-6
 - boîte de dialogue (illustration), 7-7
 - niveau de coopération, 7-9
 - option Compacter la mémoire pendant l'exécution, 7-9
 - option Vérifier l'espace disque disponible au démarrage, 7-9
 - stockage, 7-31
 - Préférences diverses, boîte de dialogue, 7-23
 - Préférences, option, 7-1
 - Presse-Papiers, 2-13
 - priorité des tâches, 26-7 à 26-23
 - à l'aide des fonctions Attendre, 26-8
 - conditions de compétition, 26-16 à 26-17
 - conseils d'utilisation des systèmes d'exécution, 26-22 à 26-23
 - dans la configuration du VI, 26-8
 - du thread de l'interface utilisateur et du système d'exécution à un seul thread, 26-9
 - exécution ré-entrante
 - utilisation d'un VI de stockage qui n'est pas sensé partager ses données (exemple), 26-15 à 26-16
 - fonctions d'occurrence, 26-21
 - fonctions de synchronisation, 26-21
 - sémaphores, 26-19 à 26-21
 - synchronisation de l'accès aux variables globales et locales, et aux ressources externes, 26-17
 - système d'exécution pour chaque niveau de priorité, 26-10 à 26-11
 - variables globales fonctionnelles, 26-18
 - VI de file d'attente, 26-22
 - VI de notification, 26-22
 - VI de rendez-vous, 26-22
 - Priorité, option, 6-3
 - priorités, en multitâche. *Voir* fonctionnement multitâche
 - problèmes de plate-forme et compatibilités, questions concernant, B-6 à B-7
 - profil de performances, 28-1 à 28-6
 - programmation modulaire, 1-2
 - programmation par flux de données et buffers de données, 28-16 à 28-18
 - programmation séquentielle
 - définition, 1-6
 - propriété Tous les VI en mémoire, 21-9
 - propriétés et méthodes
 - manipulation
 - d'application, 21-9
 - de VIs, 21-7 à 21-9
- ## Q
- questions concernant LabVIEW
 - graphes et graphes déroulants, B-1 à B-4
 - impression, B-8, B-10
 - problèmes de plate-forme et compatibilités, B-6 à B-7
 - questions diverses, B-11 à B-18
- ## R
- Rafraîchissement progressif, option, 7-11
 - rampe de couleurs, 9-24 à 9-25
 - rangées, redimensionnement, 11-8

- recherche
 - boîte de dialogue, 3-23 à 3-28
 - résultats, 3-27 à 3-28
 - texte, 3-25 à 3-26
 - utilitaire, 3-22 à 3-28
 - variables globales, locales et Attribute Nodes, 3-29
 - VI et autres objets, 3-23 à 3-24
- Rechercher le terminal, option, 8-3
- recupération de données contrôlée par programme, 4-7 à 4-9
 - à l'aide de fonctions d'E/S de fichier, 4-8
 - accès aux bases de données, 4-7 à 4-8
 - terminaux du halo pour accéder aux données, 4-7 à 4-8
- redimensionnement
 - ajout d'espace de travail, 2-24
 - clusters, 14-24
 - objets, 2-22 à 2-24
 - tableaux, 14-14
 - tables, rangées et colonnes, 11-8
- Référence en ligne, 1-10
- références d'application et de VI
 - création, 21-4
 - fonctions du VI Serveur, 21-3
 - transparence du réseau, 21-3
 - utilisation avec des noeuds de propriété et de méthode, 21-4 à 21-7
- Refnum Automation, 12-5
- RefNum d'occurrence, 12-4
- RefNum de connexion réseau, 12-5
- RefNum de fichier d'enregistrement de données, 12-4
- RefNum de fichier de structure d'octet, 12-4
- RefNum de file d'attente, 12-6
- RefNum de notificateur, 12-5
- RefNum de périphérique, 12-4
- RefNum de rendez-vous, 12-6
- RefNum de sémaphore, 12-6
- Refnum de VI ou d'application
 - description, 12-5
 - RefNum VISA, 12-5
 - RefNums de VI ou d'application
 - refnums de VI de type strict, 21-10 à 21-12
 - exemple de refnums de VI de type strict, 21-11
 - registres à décalage
 - définition, 19-4, 19-9
 - intérêt et utilisation, 19-9
 - terminaux de gauche et de droite, 19-10
 - réinitialisation des informations de l'historique, 27-12
 - Remplacer, option, 8-4
 - repères d'échelle, 9-15 à 9-19
 - changer les limites d'une échelle, 9-15 à 9-16
 - mode Repères arbitraires, 9-17
 - mode Uniforme, 9-17
 - non uniforme
 - déplacer, 9-17
 - sélectionner une distribution non uniforme, 9-16 à 9-18
 - supprimer, 9-17
 - répertoire par défaut, spécification du chemin, 7-4
 - répertoire temporaire, spécification du chemin, 7-4
 - répertoire vi.lib, 2-30
 - répertoires
 - spécification du chemin, 7-3 à 7-4
 - Chemin du recherche du VI, 7-4 à 7-6
 - répertoire de bibliothèque, 7-4
 - répertoire par défaut, 7-4
 - répertoire temporaire, 7-4
 - représentation à virgule flottante, 9-5
 - représentation des valeurs numériques
 - constantes, 17-7
 - sélection, 9-5
 - types de représentation disponibles, 9-5

- représentation en double précision (DBL) à 64 bits, 9-5
- représentation en précision étendue (EXT), 9-5
- représentation en simple précision (SGL) à 32 bits, 9-5
- représentation en simple précision, 32 bits (SGL), 9-5
- représentations d'entiers signés et non signés
 - longs (32 bits), 9-5
 - mot (17 bits), 9-5
 - octet (8 bits), 9-5
- ressources externes, synchronisation de l'accès, 26-16 à 26-17
- Retour, touche
 - arrêt de la saisie de texte, 7-10
- Revenir, option, 2-29
- Rotation de 90 Degrés, option, 3-6

S

- sélection d'objets, 2-7
- Sélectionner un élément dans le menu, 23-2
- Sélectionner une classe de VI Serveur
 - Parcourir, 21-11
 - Virtual Instrument, 21-4
 - Vis de type strict, 21-11
- sémaphores. *Voir aussi* fonctions de synchronisation, 26-19
 - création, 26-19
 - définition, 26-19
 - destruction, 26-20
 - exemple, 26-21
 - libération, 26-20
 - utilisation, 26-19
- sonde
 - Sonde automatique pendant l'exécution, sélection, 7-14
- Sonde, outil, 2-5
- sortie, 1-5

- souris
 - bouton droit pour accéder aux menus locaux, 1-4
 - déplacement entre des sous-diagrammes, 19-20
 - désactiver les interruptions de la souris, B-18
 - symbole du câblage des diagrammes, 18-1
- sous-diagrammes
 - ajout, 19-21
 - définition, 19-2
 - déplacement entre, 19-20
 - imbrication, 17-11
 - suppression, 19-22
- sous-menu Echelle des X, 15-13, 15-37
- sous-menu Echelle des Y, 15-13, 15-37
- sous-menu Opérations sur les données
 - commande Prendre la valeur actuelle par défaut, 14-22
 - commande Vider le tableau, 14-13
 - option Coller des données, 11-9
 - option Commencer la sélection, 15-30
 - option Copier des données, 11-9
 - option Couper des données, 11-9, 14-13
 - option Mode de mise à jour, 15-25
 - option Rafraîchissement progressif, 15-13
 - option Réinitialiser les valeurs par défaut, 14-22
 - option Terminer la sélection, 15-30
 - option Visualiser le dernier élément, 14-14
- options
 - Commencer la sélection, 14-15
 - Terminer la sélection, 14-15
- sous-menu Visualiser
 - option Barre de défilement, 11-3
 - option Texte booléen, 10-4
- sous-option Ignorer la casse, option Mode du clavier, 13-6

- sous-option Respecter la casse, option Mode du clavier, 13-6
- sous-palette ActiveX, 16-1
- Soustraire des fonctions
 - icône, 1-5
- sous-VIs
 - connexions obligatoires, recommandées et facultatives, 3-9
 - considérations de performances, 28-11
 - création à partir d'un VI, 3-2 à 3-10
 - création à partir d'une sélection, 3-11 à 3-15
 - étiquettes non modifiables (remarque), 2-18
 - impression, 5-1
 - impression de la liste des sous-VIs, 5-7
 - modification des descriptions (remarque), 2-27
 - réutilisation de la mémoire des données, 28-22
- Sous-VIs non ouverts, 4-15
- squelette du cluster, 14-21
- Stop, bouton
 - masquer, 6-5
- structures
 - considérations d'efficacité, 28-32 à 28-40
 - éviter les types de données compliqués, 28-34 à 28-35
 - table de chaînes de caractères globale statique, 28-39
 - table globale de type de données mélangés, 28-35 à 28-39
 - définition, 1-5
 - placement et dimensionnement sur des diagrammes, 19-5
 - problèmes de câblage
 - affectation de plusieurs valeurs à des variables locales de séquence, 19-22
 - câblage à partir de plusieurs cadres d'une structure Séquence, 19-24
 - chevauchement de tunnels, 19-23
 - suppression de structures sans suppression du contenu, 19-26
 - tunnels, 17-11
 - vue d'ensemble, 19-18
- structures Condition
 - ajout de sous-diagrammes, 19-21
 - déplacement entre des sous-diagrammes, 19-20
 - désactivation de sections de diagrammes, 4-32
 - édition, 19-20
 - icône, 19-13
 - intérêt et utilisation, 19-15 à 19-18
 - problèmes de câblage, 19-23
 - suppression de sous-diagrammes, 19-22
 - valeurs hors gamme (remarque), 19-15
 - vue d'ensemble, 19-12
- structures de données efficaces. *Voir* performances
- structures Séquence
 - ajout de sous-diagrammes, 19-21
 - but et utilisation, 17-11
 - déplacement entre des sous-diagrammes, 19-20
 - édition des sous-diagrammes, 19-20
 - intérêt et utilisation, 19-18 à 19-20
 - problèmes de câblage
 - affectation de plusieurs valeurs à des variables locales de séquence, 19-22
 - câblage à partir de plusieurs cadres d'une structure Séquence, 19-24
 - câblage par dessous plutôt qu'à travers des structures, 19-25
 - chevauchement de tunnels, 19-23
 - suppression de sous-diagrammes, 19-22
 - variable locale de séquence, 19-19
 - vue d'ensemble, 19-18

- suppression d'objets
 - connexions, 3-10
 - curseurs du graphe, 15-29 à 15-33
 - méthodes, 2-14
 - sous-diagrammes, 19-22
 - suppression de structures sans
 - suppression du contenu, 19-26
 - suppression de structures sans suppression du contenu, 19-26
 - suppression du contenu d'une bibliothèque de VIs, 2-32
 - supprimer
 - des fils, 18-6 à 18-8
 - supprimer des objets
 - Supprimer la boucle While, option, 2-14
 - symbole Chemin, 12-2
 - symbole Pas un chemin, 12-2
 - système d'exécution du G, 26-1
 - application à plusieurs threads et systèmes d'exécution multiples, 26-4 à 26-7
 - gestion de l'interface utilisateur dans une application à un seul thread, 26-4
 - multithreading, 26-2
 - noeuds synchrones de blocage, 26-7
 - présentation, 26-3
 - présentation du fonctionnement multitâche, 26-1 à 26-2
 - priorité des tâches, 26-7 à 26-23
 - à l'aide des fonctions Attendre, 26-8
 - conditions de compétition, 26-16 à 26-17
 - conseils d'utilisation des systèmes d'exécution, 26-22 à 26-23
 - dans la configuration du VI, 26-8 à 26-12
 - du thread de l'interface utilisateur et du système d'exécution à un seul thread, 26-9
 - fonctions de synchronisation, 26-21
 - niveau de priorité de type
 - sous-programme, 26-11 à 26-12
 - sémaphores, 26-19 à 26-21
 - synchronisation de l'accès aux variables globales et locales, et aux ressources externes, 26-17
 - système d'exécution pour chaque niveau de priorité, 26-10 à 26-11
 - variables globales fonctionnelles, 26-16 à 26-18
 - vue d'ensemble de l'exécution ré-entrante, 26-16
 - système d'exécution par acquisition de données, 26-6
 - système d'exécution Standard, 26-6
 - système de développement LabVIEW
 - aide, 1-7 à 1-10
 - enregistrement des VIs, 2-28
 - fenêtre d'aide, 1-7
 - menus, 2-1 à 2-3
 - outils, 2-4 à 2-5
 - utilisation, 2-5
 - référence en ligne, 1-10
 - système Interface utilisateur
 - priorité des tâches, 26-7 à 26-10
 - vue d'ensemble
- ## T
- tableau de points, 15-4
 - tableau vierge
 - composants de, 14-6
 - non identifié, 14-12
 - placement sur la face-avant, 14-6
 - tableaux, 14-2 à 14-20
 - à deux dimensions, 14-3
 - à une dimension, 14-3
 - définition, 14-2
 - définition de la taille et des valeurs par défaut, 14-11 à 14-13
 - déplacement, 14-14
 - descripteurs de type, A-11
 - description, 14-2 à 14-5

- données aplaties, A-14
- exécution, 14-11 à 14-17
- exemples, 14-2 à 14-5
- formats d'enregistrement des données, A-4
- indices, 14-2 à 14-5
- quand utiliser, 14-5
- recherche de la taille du tableau, 14-14
- redimensionnement, 14-14
- sélection des cellules, 14-15
- tableaux de LabVIEW comparés à ceux d'autres systèmes, 14-17 à 14-20
- tableaux vides, 14-12
- trois dimensions, 14-11
- tableaux à deux dimensions, 14-3, 14-10
- tableaux à trois dimensions, 14-11
- tableaux à une dimension, 14-3, 14-10
- tableaux vides, 14-12
- tables, 11-7 à 11-10
 - définition, 11-7
 - entrer et sélectionner des données, 11-8 à 11-10
 - illustration, 11-7
 - redimensionnement des tables, des rangées et des colonnes, 11-8
- terminaisons de destination, 17-2
- terminal conditionnel, 19-4
- terminal d'itération, 19-3
- terminal de comptage, boucle For, 19-3
- terminal de connecteur
 - affectation de terminaux aux commandes et indicateurs, 3-7 à 3-8
 - définition, 3-4 à 3-5
 - modification, 3-6
 - sélection, 3-6
- terminaux
 - affichage, 17-1
 - définition, 17-1
 - destination, 1-5
 - placement à l'intérieur des boucles For et While, 19-6
 - ports d'entrée et de sortie, 1-5
 - source, 1-5
 - suppression, 1-5
 - terminal conditionnel, 19-4
 - terminal d'itération, 19-3
 - terminal de comptage, 19-3
 - terminaux d'entrée de noeud, 17-1
 - terminaux d'indicateur de la face-avant, 17-2
 - terminaux de registre à décalage, 19-9
 - types de, 17-1
- terminaux de connecteur
 - confirmation des connexions, 3-10
 - fenêtre Hiérarchie, 3-15
 - suppression de connexions, 3-10
- terminaux de destination, 17-2
 - définition, 1-5
 - fils de liaison, 1-5
- terminaux source, 17-1
 - définition, 1-5
 - fils de liaison, 1-5
- texte
 - étiquettes de texte sur les échelles numériques, 9-18
 - étiquettes min et max, 9-18
 - menus déroulants
 - ajouter du texte, 13-11
 - changer du texte, 13-12
- Texte, outil, 2-5, 2-16
- touche <Alt>
 - amener le diagramme d'un sous-VI au premier plan, 4-22
 - cloner un noeud existant, 22-4
 - création d'un nouveau repère d'échelle, 9-17
 - raccourci clavier du bouton Exécution détaillée, 4-21
 - raccourci clavier du bouton Exécution non détaillée, 4-21
 - raccourci clavier du bouton Sortie, 4-21

- touche <commande>
 - raccourci clavier du bouton Exécution détaillée, 4-21
 - raccourci clavier du bouton Exécution non détaillée, 4-21
 - raccourci clavier du bouton Sortie, 4-21
- touche <Ctrl>
 - amener le diagramme d'un sous-VI au premier plan, 4-22
 - cloner un noeud existant, 22-4
 - création d'un nouveau repère d'échelle, 9-17
 - raccourci clavier du bouton Exécution détaillée, 4-21
 - raccourci clavier du bouton Exécution non détaillée, 4-21
 - raccourci clavier du bouton Sortie, 4-21
- touche <Entrée>
 - ajouter du texte aux menus déroulants, 13-10
- touche <Maj-Entrée>
 - créer des étiquettes d'échelle de texte, 9-18
 - parcourir les étiquettes d'échelle de texte, 9-19
- touche <Maj-Retour>
 - créer des étiquettes d'échelle de texte, 9-18
- touche <meta>
 - amener le diagramme d'un sous-VI au premier plan, 4-22
 - cloner un noeud existant, 22-4
 - création d'un nouveau repère d'échelle, 9-17
 - raccourci clavier du bouton Exécution détaillée, 4-21
 - raccourci clavier du bouton Exécution non détaillée, 4-21
 - raccourci clavier du bouton Sortie, 4-21
- touche <option>
 - amener le diagramme d'un sous-VI au premier plan, 4-22
 - cloner un noeud existant, 22-4
 - création d'un nouveau repère d'échelle, 9-17
- touche <retour arrière>
 - pour supprimer un fil, 18-6
- touche <Suppr>
 - suppression de fils, 18-6
- tracés, 15-1
- tracés empilés, 15-27
- tracés superposés, 15-27
- tunnels
 - définition, 17-11, 18-2
 - déplacement, 18-7
 - problèmes de câblage, 19-23
- type de données
 - appel de fonctions qui attendent des types de données non LabVIEW, 25-8
 - chaînes de caractères, 25-7
 - impressions du type de données des commandes, 5-7
 - numériques, 25-6
 - tableaux, 25-6
 - vide, 25-6
- type de données chaîne de caractères, 25-7
- type de données numérique, 25-6
- types de données
 - considérations de mémoire
 - éviter de redimensionner constamment les données, 28-26
 - générer des données du type voulu, 28-25 à 28-26
 - types de données compatibles, 28-24 à 28-26
 - éléments des listes déroulantes, 13-3
 - enregistrement de quantités physiques (remarque), A-11
 - éviter les types de données compliqués dans les structures, 28-34 à 28-35

- exemples, A-10
- non numériques (tableau), A-8
- numériques scalaires (tableau), A-8
- types de données des graphes XY, 15-4
- types de données discontinus, A-12
- types de données énumérés, changer au moyen d'un programme, B-18
- types de données non numériques (tableau), A-8
- types de données scalaires
 - données aplaties, A-13
 - numériques et non numériques (tableaux), A-8 à A-10

U

- unités, 9-25 à 9-33
 - entrée des unités, 9-30
 - menu local, 9-26
 - types
 - autres unités (tableau), 9-29
 - unités CGS (tableau), 9-29
 - unités de base (tableau), 9-27
 - unités dérivées à noms spéciaux (tableau), 9-27 à 9-28
 - unités supplémentaires en usage avec des unités SI (tableau), 9-28
 - unités et contrôle de type strict, 9-31 à 9-32
 - unités polymorphes, 9-32
 - visualisation de l'étiquette d'unité, 9-25
- unités et contrôle de type strict, 9-31 à 9-32
- unités polymorphes, 9-32
- usage de mémoire
 - charger dynamiquement des VIs, B-12
- utilisation de la mémoire, 28-13 à 28-31
 - composantes des VIs, 28-15
 - concepts de base, 28-14 à 28-18
 - contrôle, 28-18 à 28-19
 - faces-avant, 28-20 à 28-22
 - généralités, 28-13

- libération par LabVIEW, 28-23 à 28-24
- mémoire virtuelle, 28-14
- programmation par flux de données et buffers de données, 28-16 à 28-18
- règles générales pour une amélioration, 28-19 à 28-20
- réutilisation du buffer d'entrée par la sortie, 28-24
- réutilisation par les sous-VIs de la mémoire des données, 28-22
- variables globales, 28-23
- variables locales, 28-22

V

- variable locale de séquence
 - ajout, 19-19
- variables globales
 - considérations de mémoire, 28-23
 - création, 23-1
 - définition, 23-1
 - exemple, 23-2
 - fonctionnelles, 26-18
 - intérêt et utilisation, 23-1 à 23-3
 - placement des VIs, 23-3
 - sélection, 23-2
 - synchronisation de l'accès, 26-16 à 26-18
- variables locales, 23-4 à 23-5
 - création, 23-4
 - incapacité à réutiliser la mémoire des données, 28-22
 - plusieurs variables locales, 23-5
 - synchronisation de l'accès, 26-16
- variables locales de séquence
 - affectation de plusieurs valeurs à, 19-22
- Vérifier l'espace disque disponible au démarrage, option, 7-9
- Verrouiller le texte au centre, option, 10-3
- VI "Acquérir un sémaphore" (Acquire Semaphore.vi), 26-19, 26-20
- VI "AESend Print Document", 5-1

- VI “Créer un sémaphore” (Create Semaphore.vi), 26-19
- VI “Détruire un sémaphore” (Destroy Semaphore.vi), 26-20
- VI “Exec. système”, 5-1
- VI “Libérer un sémaphore” (Release Semaphore.vi), 26-20
- VI Serveur, 21-1 à 21-12
 - fonctions et utilisation, 21-1 à 21-3
 - manipulation des méthodes et propriétés d’application, 21-9
 - d’application et de VIs, 21-10
 - de VIs, 21-7
 - références d’application et de VI, 21-4
 - création, 21-4
 - utilisation des noeuds de propriété et de méthode, 21-4 à 21-7
 - refnums de VI de type strict
 - exemple, 21-11
- VIs
 - caractéristiques, 1-2
 - chargement
 - questions concernant, B-13
 - comparaison avec les fonctions dans d’autres langages, 1-1 à 1-2
 - définition, 1-1
 - enregistrement, 2-28 à 2-32
 - bibliothèques de VIs, 2-30
 - dans des répertoires autres que vi.lib (avertissement), 2-30
 - modification des VIs après enregistrement (avertissement), 2-29
 - option Revenir, 2-29
 - options, 2-28
 - éviter la fermeture de VIs, 6-4
 - fonctionnement
 - éviter la fermeture de VIs en exécution par un utilisateur, 6-4
 - lancer automatiquement, B-15
 - modification
 - alignement d’objets, 2-11
 - copie d’objets, 2-13
 - création d’icônes, 3-2
 - création de descriptions, 2-26
 - définition d’un terminal de connecteur, 3-4 à 3-5
 - déplacement d’objets, 2-9 à 2-10
 - option Placer en arrière-plan, 2-11
 - option Placer en premier plan, 2-11
 - options
 - Placer derrière, 2-11
 - Placer devant, 2-10
 - documentation des VIs, 2-27
 - étiquettes
 - création, 2-16 à 2-18
 - modification des attributs de texte, 2-18 à 2-22
 - étiquettes des objets, 2-14
 - redimensionnement d’objets, 2-22 à 2-24
 - redimensionnement des étiquettes, 2-23
 - répartition d’objets, 2-12
 - sélection d’objets, 2-7 à 2-8
 - suppression d’objets, 2-14
 - portabilité, 29-1 à 29-3
 - différences de résolution et de fonte, 29-3 à 29-5
 - différences entre les images, 29-5
 - différences entre les séparateurs, 29-2
 - généralités, 29-1
 - VIs non portables, 29-1 à 29-2
 - VIs AppleEvent
 - pour l’exécution d’autres applications, 25-1
 - VIs de file d’attente, 26-22
 - VIs de notification, 26-22
 - VIs de rendez-vous, 26-22
 - VIs exécutables, 27-3 à 27-4

- correction des erreurs de dépassement de gamme des VIs invalides, 4-14
 - localisation des erreurs, 4-10
 - messages d'erreur (tableau), 4-11 à 4-13
 - raisons courantes pour les VIs invalides, 4-10
 - réparation, 4-9 à 4-13
- VIs Port série, 5-1
- VIs. *Voir aussi* applications, gestion, 5-1
- VIs. *Voir aussi* applications, gestion; sous-VIs
 - arrêt, 4-4
 - exécution, 4-1 à 4-3
 - exécution répétée, 4-4
 - impression, 5-3 à 5-16
 - autres méthodes d'impression, 5-10
 - configuration de l'impression, 5-2
 - contrôlée par programme, 5-7 à 5-10
 - fenêtre active, 5-2
 - informations concernant
 - l'historique, 5-7
 - options, 5-3 à 5-7
- Visualiser la barre d'outils, option, 6-5
- Visualiser la face-avant lors de l'appel, option, 6-2
- Visualiser la face-avant lors du chargement, option, 6-2
- Visualiser la hiérarchie des VIs, option palette Projet, 3-15
- Visualiser le connecteur, option, 3-4
- Visualiser les infos du VI, option (remarque), 2-27
 - documentation des VIs, 2-27
 - modification des descriptions (note), 2-27
- Visualiser les mises en garde par défaut dans la boîte d'erreurs, option, 7-14
- Visualiser, sous-menu
 - Etiquette, 2-17
 - palette Commandes, 8-3