

LabVIEW™ Fundamental 1

Manuel de cours

Logiciel de cours version 2011
Édition d'août 2011
Référence 325290C-0114

Copyright

© 1993–2011 National Instruments Corporation. Tous droits réservés.

Conformément à la réglementation applicable en matière de droits d'auteur, cette publication ne peut pas être reproduite ni transmise sous une forme quelconque, que ce soit par voie électronique ou mécanique, notamment par photocopie, enregistrement ou stockage dans un système permettant la récupération d'informations, ni traduite, en tout ou partie, sans le consentement préalable et écrit de National Instruments Corporation.

National Instruments respecte les droits de propriété intellectuelle appartenant à des tiers et nous demandons aux utilisateurs de nos produits de les respecter également. Les logiciels NI sont protégés par la réglementation applicable en matière de droits d'auteur et de propriété intellectuelle. Lorsque des logiciels NI peuvent être utilisés pour reproduire des logiciels ou autre matériel appartenant à des tiers, vous ne pouvez utiliser les logiciels NI à cette fin que pour autant que cette reproduction est permise par les termes du contrat de licence applicable auxdits logiciels ou matériel et par la réglementation en vigueur.

For components used in USI (Xerces C++, ICU, HDF5, b64, Stingray, and STLport), the following copyright stipulations apply. For a listing of the conditions and disclaimers, refer to either the `USICopyrights.chm` or the *Copyrights* topic in your software.

Xerces C++. This product includes software that was developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright 1999 The Apache Software Foundation. All rights reserved.

ICU. Copyright 1995–2009 International Business Machines Corporation and others. All rights reserved.

HDF5. NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities

Copyright 1998, 1999, 2000, 2001, 2003 by the Board of Trustees of the University of Illinois. All rights reserved.

b64. Copyright © 2004–2006, Matthew Wilson and Synesis Software. All Rights Reserved.

Stingray. This software includes Stingray software developed by the Rogue Wave Software division of Quovadx, Inc.

Copyright 1995–2006, Quovadx, Inc. All Rights Reserved.

STLport. Copyright 1999–2003 Boris Fomitchev

Marques

CVI, LabVIEW, National Instruments, NI, ni.com, le logo de la société National Instruments et le logo de l'Aigle sont des marques de National Instruments Corporation. Veuillez consulter la rubrique *Trademark Information* sur ni.com/trademarks pour d'autres marques de National Instruments.

The mark LabWindows is used under a license from Microsoft Corporation. Windows is a registered trademark of Microsoft Corporation in the United States and other countries. Les autres noms de produits et de sociétés mentionnés aux présentes sont les marques ou les noms de leurs propriétaires respectifs.

Les membres du programme "National Instruments Alliance Partner Program" sont des entités professionnelles indépendantes de National Instruments et aucune relation d'agence, de partenariat ou « joint-venture » n'existe entre ces entités et National Instruments.

Brevets

Pour la liste des brevets protégeant les produits/technologies National Instruments, veuillez vous référer, selon le cas : à la rubrique **Aide»Brevets** de votre logiciel, au fichier `patents.txt` sur votre média, ou à *National Instruments Patent Notice* sur ni.com/patents.

Filiales francophones

National Instruments
France
2 rue Hennape
92735 Nanterre Cedex

National Instruments
Suisse
Sonnenbergstr. 53
CH-5408 Ennetbaden

National Instruments
Belgium nv
Ikaroslaan 13
B-1930 Zaventem

National Instruments
Canada
1 Holiday Street
East Tower, Suite 501
Point-Claire, Québec H9R 5N3

Support

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

Site FTP :
ftp.ni.com

Adresse Web :
france.ni.com
ni.com/support
suisse.ni.com
belgique.ni.com
canada.ni.com

Téléphone :

France	Tél. : 01 57 66 24 24	Fax : 01 57 66 24 14	
Suisse	Tél. : 056 2005151	Fax : 056 200 51 55	
Belgique	Tél. : 02 757 0020	Fax : 02 757 03 11	Tél. : 4050120 (Luxembourg)
Canada (Québec)	Tél. : 450 510 3055	Fax : 450 510 3056	

Filiales internationales

Visitez ni.com/niglobal pour accéder aux sites Web des branches. Vous y trouverez les informations les plus à jour pour contacter le support technique par téléphone ou e-mail, ainsi que le calendrier des événements.

Siège social de National Instruments

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

Pour plus d'informations de support, consultez l'annexe *Informations et ressources supplémentaires*. Si vous souhaitez formuler des commentaires sur la documentation National Instruments, reportez-vous au site Web de National Instruments sur ni.com/frinfo et entrez l'info-code `feedback`.

Sommaire

Guide du stagiaire

A. Certification NI	vii
B. Description du cours	viii
C. Configuration système et matériel requis pour commencer	ix
D. Installation du logiciel de cours	x
E. Objectifs du cours	xi
F. Conventions utilisées dans ce cours.....	xii

Leçon 1

Configuration du matériel

A. Matériel DAQ	1-2
B. Utilisation de logiciels DAQ.....	1-9
C. Contrôle d'instruments	1-13
D. GPIB	1-13
E. Communication port série.....	1-14
F. Utilisation de logiciels de contrôle d'instrument.....	1-17
G. Projet de cours	1-19

Leçon 2

Navigation dans LabVIEW

A. Instruments virtuels (VIs)	2-2
B. Composantes d'un VI	2-2
C. Démarrage d'un VI	2-4
D. Explorateur de projet	2-9
E. Face-avant	2-14
F. Diagramme.....	2-22
G. Recherche de commandes, de VIs et de fonctions.....	2-32
H. Sélection d'un outil	2-36
I. Flux de données	2-43
J. Création d'un VI simple	2-45

Leçon 3

Identification des problèmes et mise au point des VIs

A. Utilitaires d'aide LabVIEW	3-2
B. Correction des VIs brisés	3-5
C. Techniques de mise au point.....	3-7
D. Données non définies ou inattendues.....	3-14
E. Détection et gestion des erreurs	3-15

Leçon 4**Implémentation d'un VI**

A. Conception de la face-avant.....	4-2
B. Types de données LabVIEW	4-10
C. Documentation du code	4-20
D. Boucles While.....	4-23
E. Boucles For	4-27
F. Cadencement d'un VI	4-32
G. Transfert de données itératif	4-33
H. Tracé des données	4-37
I. Structures Condition	4-44

Leçon 5**Regroupement des données**

A. Tableaux.....	5-2
B. Clusters	5-8
C. Définitions de type.....	5-14

Leçon 6**Gestion des ressources**

A. Introduction aux E/S sur fichiers	6-2
B. Introduction aux E/S sur fichiers de haut niveau	6-3
C. Introduction aux E/S sur fichiers de bas niveau	6-4
D. Programmation DAQ.....	6-6
E. Programmation pour le contrôle d'instruments	6-8
F. Utilisation de drivers d'instruments.....	6-11

Leçon 7**Développement d'applications modulaires**

A. Introduction à la modularité.....	7-2
B. Construction de l'icône et du connecteur	7-4
C. Utilisation de sous-VIs.....	7-9

Leçon 8**Techniques et modèles de conception courants**

A. Utilisation d'une programmation séquentielle.....	8-2
B. Utilisation d'une programmation à états.....	8-4
C. Machines à états.....	8-5
D. Utilisation du parallélisme	8-14

Leçon 9**Utilisation de variables**

A. Parallélisme.....	9-2
B. Variables	9-4
C. Variables globales fonctionnelles	9-15
D. Situation de compétition	9-18

Annexe A**Principes de base des mesures**

A. Utilisation de systèmes de mesure informatiques.....	A-2
B. Introduction aux concepts de mesure.....	A-3
C. Amélioration de la qualité des mesures	A-12

Annexe B**Informations et ressources supplémentaires****Glossaire****Index**

Échantillon

Regroupement des données

Il est parfois utile de regrouper des données apparentées. Utilisez des tableaux et des clusters pour regrouper des données apparentées dans LabVIEW. Les tableaux combinent des données du même type dans une structure de données, et les clusters combinent des données de types différents dans une structure de données. Utilisez des définitions de type pour définir des tableaux et des clusters personnalisés. Cette leçon traite des tableaux, des clusters et des définitions de type, ainsi que des applications où leur utilisation peut s'avérer utile.

Sujets abordés

- A. Tableaux
- B. Clusters
- C. Définitions de type

A. Tableaux

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

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



Remarque Les indices de tableau dans LabVIEW commencent à la valeur zéro. L'indice du premier élément dans le tableau, quelle que soit sa dimension, est zéro.

Restrictions

Vous ne pouvez pas créer des tableaux de tableaux. Vous pouvez néanmoins créer un tableau de clusters où chaque cluster contient un ou plusieurs tableaux. Par ailleurs, vous ne pouvez pas créer un tableau de commandes de face-avant secondaire, de commandes onglet, de contrôles .NET ou ActiveX, de graphes déroulants ou de graphes XY multitracés. Reportez-vous à la section de cette leçon portant sur les clusters pour obtenir de plus amples informations sur les clusters.

Un exemple de tableau simple est un tableau de texte qui répertorie les douze mois de l'année. LabVIEW le représente comme un tableau de chaînes 1D à douze éléments.

Les éléments d'un tableau sont ordonnés. Un tableau utilise un indice pour vous permettre d'accéder facilement à un élément particulier. L'indice est basé sur zéro, ce qui signifie qu'il se trouve dans la gamme de 0 à $n - 1$, où n est le nombre d'éléments du tableau. Par exemple, pour les douze mois de l'année, $n = 12$ et l'indice va de 0 à 11. Mars étant le troisième mois, son indice est 2.

La figure 5-1 présente un exemple de tableau de numériques. Le premier élément visible dans le tableau (3, 00) est à l'indice 1, et le deuxième élément (1, 00) est à l'indice 2. L'élément à l'indice 0 n'est pas visible sur cette figure car l'élément 1 est sélectionné dans l'afficheur d'indice. L'élément sélectionné dans l'afficheur d'indice fait toujours référence à l'élément affiché dans le coin supérieur gauche de l'afficheur d'éléments.

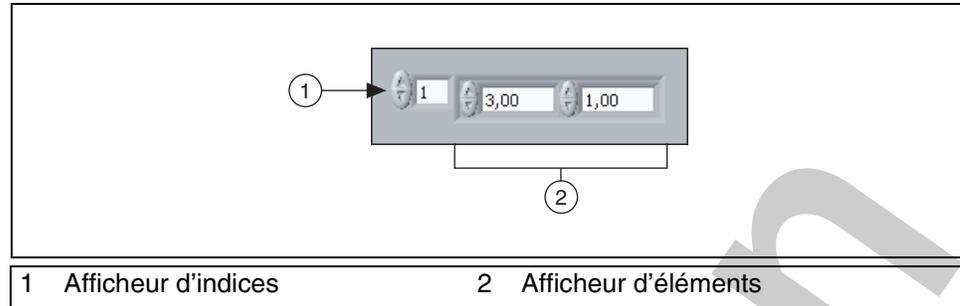


Figure 5-1. Commande tableau de numériques

Création de commandes et d'indicateurs tableau

Créez une commande ou un indicateur tableau sur la face-avant en y plaçant un tableau vide, illustré dans la figure suivante, et en y faisant glisser un élément ou un objet de données, lequel peut être une commande ou un indicateur numérique, booléen, chaîne, chemin, refnum ou cluster.

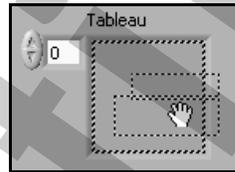


Figure 5-2. Placement d'une commande numérique dans un tableau vide

Il n'est pas possible de faire glisser une commande ou un indicateur non valide dans un tableau.

Avant d'utiliser un tableau dans le diagramme, vous devez insérer un objet dans le tableau vide. Sinon, le terminal tableau est noir et aucun type de données n'apparaît entre les crochets.

Tableaux à deux dimensions

Les exemples précédents utilisent des tableaux 1D. Un tableau 2D stocke des éléments dans une grille. Pour repérer un élément, vous avez besoin d'un indice de colonne et d'un indice de ligne, les deux commençant à zéro. La figure 5-3 montre un tableau 2D à 8 colonnes et 8 lignes contenant $8 \times 8 = 64$ éléments.

		Indices des colonnes							
		0	1	2	3	4	5	6	7
Indices des lignes	0								
	1								
	2								
	3								
	4								
	5								
	6								
	7								

Figure 5-3. Tableau 2D

Pour ajouter un tableau multidimensionnel sur la face-avant, effectuez un clic droit sur l’afficheur d’indice et sélectionnez **Ajouter une dimension** dans le menu local. Vous pouvez aussi redimensionner l’afficheur d’indice jusqu’à ce que vous obteniez le nombre de dimensions souhaitées.

Initialisation des tableaux

Vous pouvez initialiser un tableau ou le laisser non initialisé. Un tableau est initialisé si vous avez défini le nombre d’éléments de chaque dimension et le contenu de chaque élément. Un tableau non initialisé contient un nombre fixe de dimensions, mais pas d’éléments. La figure 5-4 représente une commande tableau 2D non initialisée. Notez que tous les éléments sont grisés. Ceci indique que le tableau n’est pas initialisé.



Figure 5-4. Tableau 2D non initialisé

Dans la figure 5-5, six éléments sont initialisés.

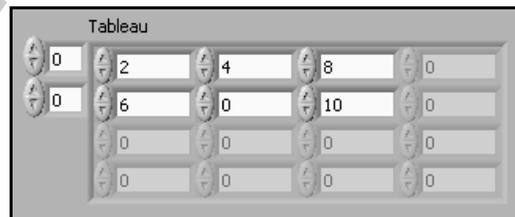


Figure 5-5. Tableau 2D initialisé avec six éléments

Dans un tableau 2D, une fois que vous avez initialisé un élément dans une rangée, les éléments de cette rangée sont initialisés et remplis par la valeur

par défaut de ce type de données. Par exemple, dans la figure 5-6, si vous entrez 4 dans l'élément de la première colonne, sur la troisième ligne, un 0 apparaît automatiquement dans les éléments de la troisième ligne des deuxième et troisième colonnes.

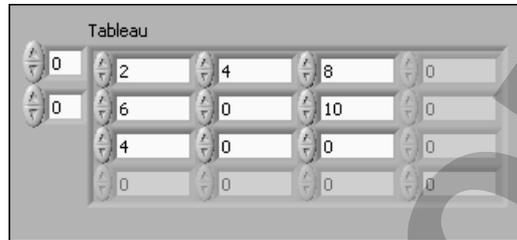


Figure 5-6. Tableau qui a été automatiquement rempli avec des zéros

Création de constantes tableau

Pour créer une constante tableau sur le diagramme, sélectionnez une constante tableau sur la palette **Fonctions**, positionnez le tableau vide sur le diagramme et placez-y une constante chaîne, une constante numérique, une constante booléenne ou une constante cluster. Vous pouvez utiliser une constante tableau pour enregistrer les données d'une constante ou pour effectuer une comparaison avec un autre tableau.

Auto-indexation des entrées de tableau



Si vous câblez un tableau à une boucle For ou une boucle While, vous pouvez associer chaque itération de la boucle à un élément de ce tableau en activant l'auto-indexation. Pour indiquer que l'auto-indexation est activée, l'image du tunnel passe d'un carré plein à l'illustration. Pour changer l'état du tunnel, effectuez un clic droit sur le tunnel et sélectionnez **Activer l'indexation** ou **Désactiver l'indexation** dans le menu local.

Tableaux en entrée

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

Dans la figure 5-7, la boucle For s'exécute un nombre de fois égal au nombre d'éléments du tableau. Normalement, si le terminal de décompte de la boucle For n'est pas câblé, la flèche d'exécution est brisée. Cependant, dans ce cas-ci, la flèche d'exécution n'est pas brisée.

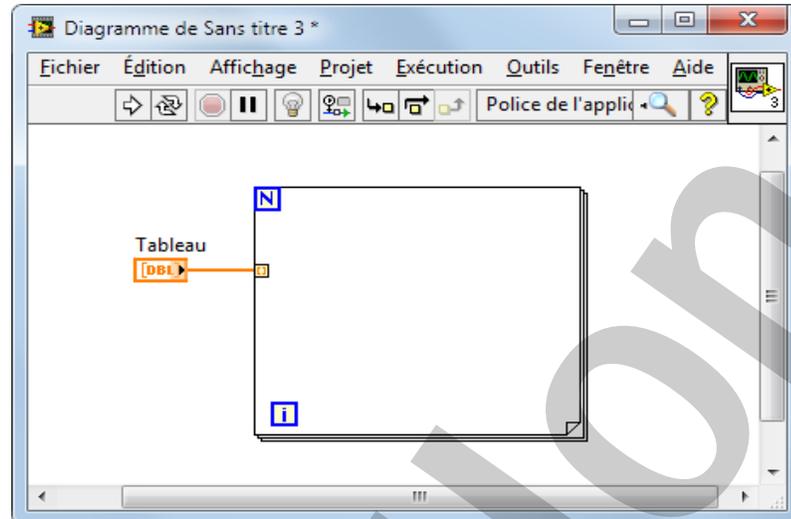


Figure 5-7. Tableau utilisé pour définir le nombre d'itérations de la boucle For

Si vous activez l'auto-indexation pour plusieurs tunnels ou si vous câblez le terminal de décompte, le nombre d'itérations sera le plus petit parmi les choix. Par exemple, si deux tableaux auto-indexés entrent dans la boucle, avec respectivement 10 et 20 éléments, et que vous câblez la valeur 15 au terminal de décompte, la boucle exécute 10 itérations en indexant tous les éléments du premier tableau mais en n'indexant que les 10 premiers éléments du deuxième tableau.

Tableaux en sortie

Lorsque vous auto-indexez le tunnel de sortie d'un tableau, le tableau en sortie reçoit un nouvel élément à chaque itération de la boucle. Ainsi, les tableaux auto-indexés en sortie ont toujours une taille égale au nombre d'itérations.

Le fil reliant le tunnel de sortie à l'indicateur tableau est plus épais et le tunnel de sortie contient des crochets représentant un tableau, comme le montre la figure 5-8.

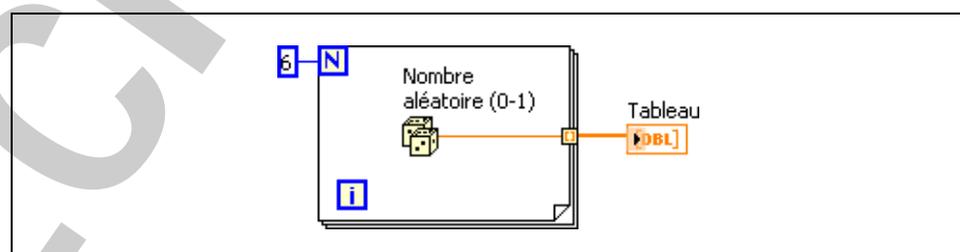


Figure 5-8. Sortie auto-indexée

Cliquez avec le bouton droit sur le tunnel sur le cadre de la boucle et sélectionnez **Activer l'indexation** ou **Désactiver l'indexation** dans le

menu local pour activer ou désactiver l'auto-indexation. L'auto-indexation pour les boucles While est désactivée par défaut.

Désactivez l'indexation automatique si seule la dernière valeur transmise par le tunnel vous intéresse.

Création de tableaux à deux dimensions

Vous pouvez utiliser deux boucles For imbriquées l'une dans l'autre pour créer un tableau à deux dimensions. La boucle For extérieure crée les éléments de ligne tandis que la boucle intérieure crée les éléments de colonne, comme le montre la figure 5-9.

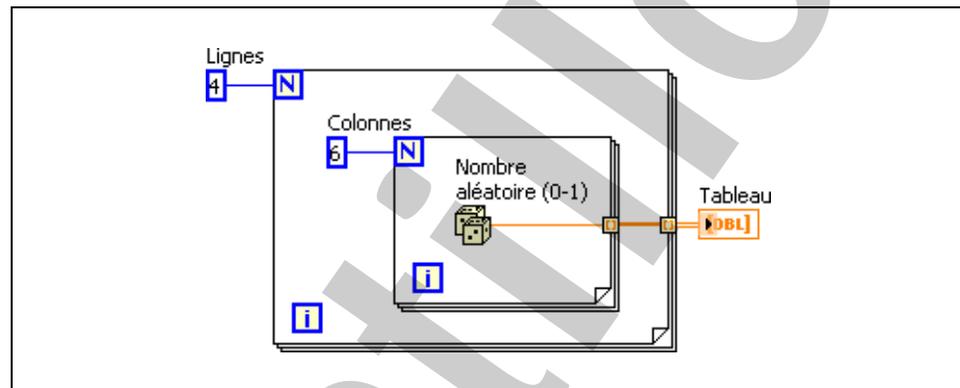


Figure 5-9. Création d'un tableau 2D

Pour mettre en pratique les concepts traités dans cette section, faites l'exercice 5-1.

B. Clusters

Les clusters regroupent des éléments de données de types différents. Le cluster d'erreur LabVIEW est un exemple de cluster ; il comprend une valeur booléenne, une valeur numérique et une chaîne. Les clusters sont semblables aux "record" et "struct" des langages de programmation textuels.

En rassemblant plusieurs éléments de données dans un cluster, vous diminuez le nombre de fils de liaison sur le diagramme tout en réduisant le nombre de terminaux de connecteurs requis par les sous-VIs. Le connecteur peut avoir un maximum de 28 terminaux. Si votre face-avant contient plus de 28 commandes et indicateurs qui seront utilisés par un autre VI, regroupez certains d'entre eux dans un cluster et affectez le cluster à un terminal du connecteur.

La plupart des clusters sur le diagramme possèdent un modèle de fil de liaison et un terminal de type de données roses. Les clusters d'erreur ont un modèle de fil et un terminal de type de données jaune foncé. Les clusters de valeurs numériques, appelés parfois des points, possèdent un modèle de fil de liaison et un terminal de type de données marron. Vous pouvez câbler les clusters numériques marron aux fonctions numériques, comme Additionner ou Racine carrée, pour réaliser la même opération simultanément sur tous les éléments du cluster.

Ordre des éléments d'un cluster

Bien que les éléments des clusters et des tableaux soient ordonnés, vous pouvez désassembler tous les éléments d'un cluster en même temps en utilisant la fonction Désassembler. Vous pouvez utiliser la fonction Désassembler par nom pour désassembler les éléments d'un cluster d'après leur nom. Si vous utilisez la fonction Désassembler par nom, tous les éléments du cluster doivent avoir une étiquette. Les clusters diffèrent également des tableaux dans la mesure où leur taille est fixe. Comme un tableau, un cluster est soit une commande soit un indicateur. Un cluster ne peut pas contenir simultanément des commandes et des indicateurs.

Création des commandes et indicateurs cluster

Créez une commande ou un indicateur cluster sur la face-avant en y plaçant un cluster vide, comme dans la figure suivante, et en y faisant glisser un élément ou un objet de données, comme une commande ou un indicateur numérique, booléen, chaîne, chemin, refnum ou cluster.

Lorsque vous déposez un cluster, vous pouvez le redimensionner en faisant glisser le curseur.

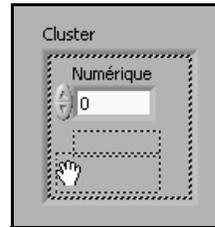


Figure 5-10. Création d'une commande cluster

La figure 5-11 représente un exemple de cluster qui comporte trois commandes : une chaîne, un commutateur booléen et un numérique. Un cluster peut être une commande ou un indicateur, mais il ne peut pas combiner des commandes et des indicateurs.

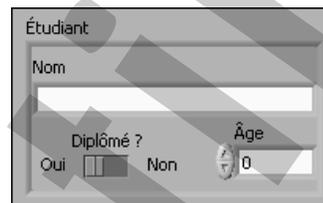


Figure 5-11. Exemple de commande cluster

Création de constantes cluster

Pour créer une constante cluster sur le diagramme, sélectionnez une constante cluster sur la palette **Fonctions**, positionnez le cluster vide sur le diagramme et mettez-y une constante chaîne, une constante numérique, une constante booléenne ou une constante cluster. Vous pouvez utiliser une constante cluster pour stocker des données de constantes ou pour effectuer une comparaison avec un autre cluster.

Si la face-avant contient une commande ou un indicateur cluster et que vous souhaitez créer une constante cluster contenant les mêmes éléments sur le diagramme, faites glisser ce cluster de la fenêtre de la face-avant jusqu'au diagramme. Une autre possibilité consiste à cliquer avec le bouton droit sur le cluster du diagramme pour sélectionner **Créer»Constante** dans le menu local.

Ordre des clusters

Les éléments de cluster possèdent un ordre logique qui n'est pas lié à leur position dans le cluster. Le premier objet que vous placez dans le cluster est l'élément 0, le deuxième est l'élément 1 et ainsi de suite. Si vous supprimez un élément, l'ordre est automatiquement ajusté. L'ordre du cluster détermine l'ordre dans lequel les éléments s'affichent comme terminaux sur

les fonctions Assembler et Désassembler du diagramme. Vous pouvez afficher et modifier l'ordre des éléments du cluster en effectuant un clic droit sur le cadre du cluster et en sélectionnant **Ordonner les commandes dans le cluster** dans le menu local.

La barre d'outils et le cluster changent alors d'aspect, comme le montre la figure 5-12.

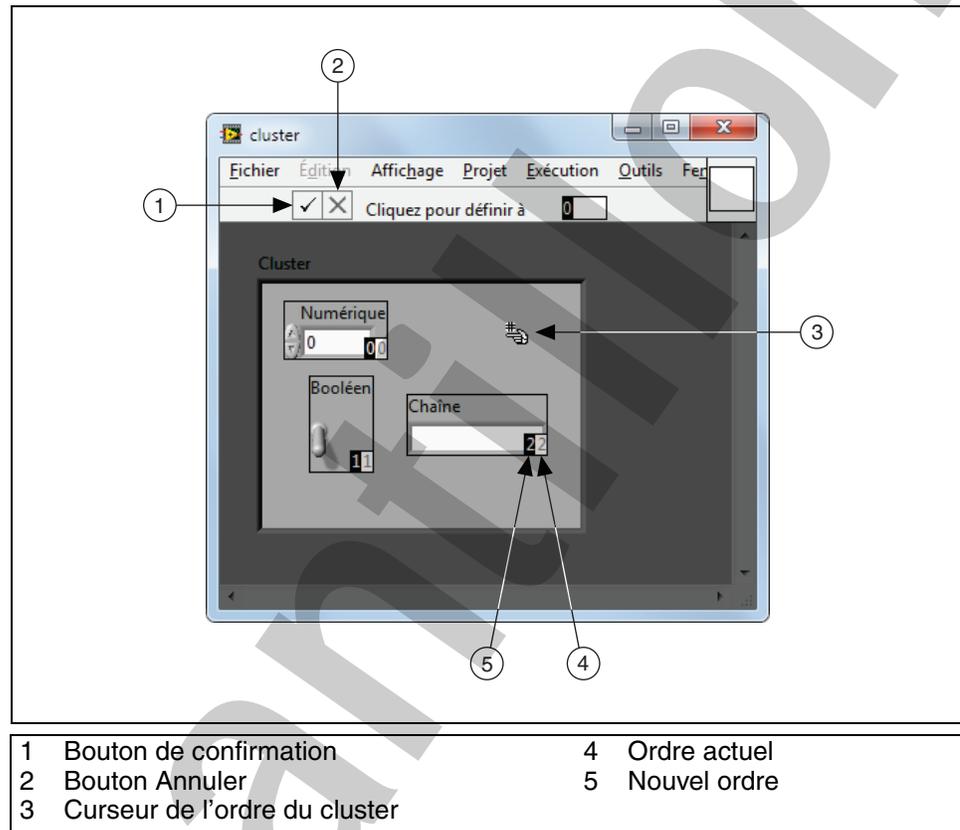


Figure 5-12. Réorganisation d'un cluster

La case blanche de chaque élément montre son emplacement actuel dans l'ordre du cluster. La case noire montre le nouvel emplacement de l'élément dans l'ordre du cluster. Pour définir l'ordre d'un élément du cluster, entrez le nouveau numéro d'ordre dans le champ **Cliquez pour définir à** et cliquez sur l'élément. L'ordre de l'élément sur lequel vous avez cliqué change et celui des autres éléments du cluster s'ajuste en conséquence. Enregistrez vos modifications en cliquant sur le bouton **Confirmer** de la barre d'outils. Pour revenir à la configuration originale, cliquez sur le bouton **Annuler**.

Utilisation des fonctions de cluster

Utilisez les fonctions de cluster pour créer et manipuler des clusters. Par exemple, vous pouvez effectuer des tâches semblables aux suivantes :

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

Utilisez la fonction Assembler pour assembler un cluster, les fonctions Assembler et Assembler par nom pour modifier un cluster et les fonctions Désassembler et Désassembler par nom pour désassembler des clusters.

Vous pouvez également déposer une fonction Assembler, Assembler par nom, Désassembler ou Désassembler par nom sur le diagramme en cliquant avec le bouton droit sur un terminal de cluster du diagramme et en sélectionnant **Palette Cluster, classe et variant** dans le menu local. Les fonctions Assembler et Désassembler contiennent automatiquement le nombre correct de terminaux. Les fonctions Assembler par nom et Désassembler par nom apparaissent avec le premier élément du cluster. Utilisez l'outil Flèche pour redimensionner les fonctions Assembler par nom et Désassembler et afficher les autres éléments du cluster.

Assemblage de clusters

Utilisez la fonction Assembler pour assembler un cluster à partir d'éléments individuels ou pour modifier les valeurs d'éléments individuels dans un cluster existant sans avoir à spécifier de nouvelles valeurs pour tous les éléments. Utilisez l'outil Flèche pour redimensionner la fonction ou cliquez avec le bouton droit sur l'entrée d'un élément et sélectionnez **Ajouter une entrée** dans le menu local.

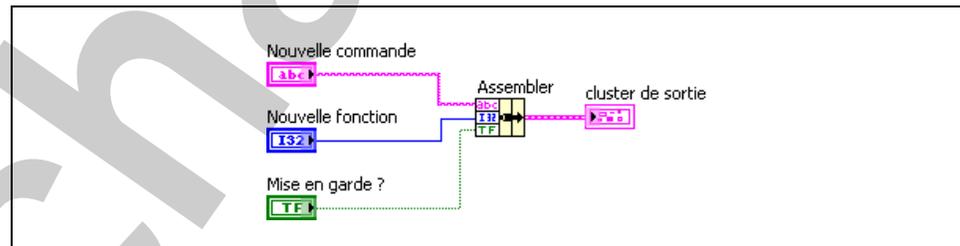


Figure 5-13. Assemblage d'un cluster sur le diagramme

Modification d'un cluster

Si vous câblez l'entrée cluster, vous pouvez vous contenter de ne câbler que les éléments que vous souhaitez changer. Par exemple, le cluster en entrée illustré à la figure 5-14 se compose de trois commandes.

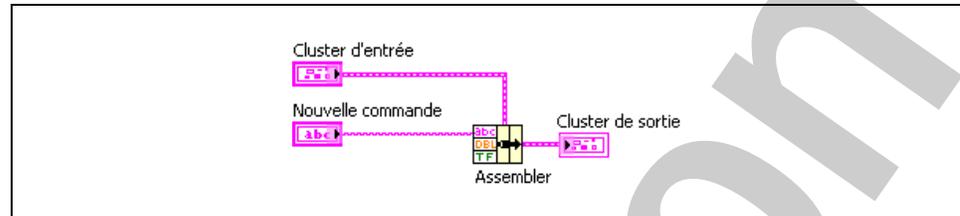


Figure 5-14. Utilisation de la fonction Assembler pour modifier un cluster

Si vous connaissez l'ordre du cluster, vous pouvez utiliser la fonction Assembler pour changer la valeur de **Commande** en câblant les éléments représentés dans la figure 5-14.

Vous pouvez aussi utiliser la fonction Assembler par nom pour accéder aux éléments nommés d'un cluster existant ou les remplacer. Assembler par nom fonctionne de la même manière que la fonction Assembler, mais au lieu de référencer les éléments du cluster dans l'ordre de celui-ci, elle les référence d'après leur étiquette liée. Vous ne pouvez accéder qu'aux éléments qui ont des étiquettes liées. Il n'est pas nécessaire que le nombre d'entrées soit identique à celui des éléments du **cluster de sortie**.

Alors que la fonction Assembler affiche toutes les parties d'un cluster, la fonction Assembler par nom peut être configurée pour n'afficher que des éléments spécifiques d'un cluster. Pour sélectionner les éléments que la fonction Assembler par nom doit afficher, utilisez l'outil Doigt pour cliquer sur un terminal de sortie de la fonction et sélectionner un élément dans le menu déroulant. Vous pouvez également cliquer avec le bouton droit sur le terminal de sortie et sélectionner l'élément dans le menu local de **Sélectionner un élément**.

Dans la figure 5-15, vous pouvez utiliser la fonction Assembler par nom pour mettre à jour les valeurs de **Commande** et de **Fonction** avec les valeurs de **Nouvelle commande** et de **Nouvelle fonction**.

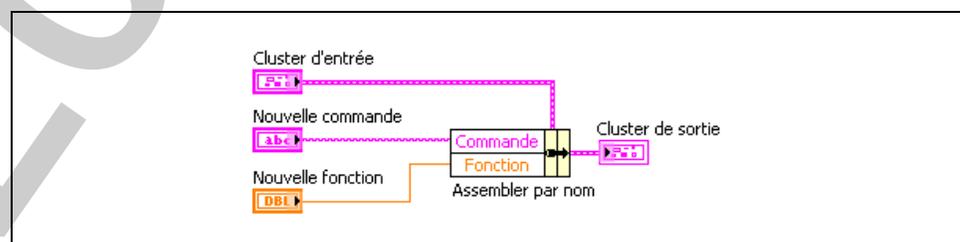


Figure 5-15. Utilisation de la fonction Assembler par nom pour modifier un cluster

Utilisez la fonction Assembler par nom pour les structures de données susceptibles de changer pendant le développement. Si vous ajoutez un nouvel élément au cluster ou si vous en modifiez l'ordre, il n'est pas nécessaire de recâbler la fonction Assembler par nom, car les noms sont encore valides.

Désassemblage de clusters

Utilisez la fonction Désassembler pour désassembler les différents éléments d'un cluster.

Utilisez la fonction Désassembler par nom pour renvoyer les éléments de cluster dont vous spécifiez le nom. Le nombre de terminaux de sortie ne dépend pas du nombre d'éléments contenus dans le cluster d'entrée.

Alors que la fonction Désassembler affiche toutes les parties d'un cluster, la fonction Désassembler par nom peut être configurée pour n'afficher que des éléments spécifiques d'un cluster. Pour sélectionner les éléments que la fonction Désassembler par nom doit afficher, utilisez l'outil Doigt pour cliquer sur un terminal de sortie de la fonction et sélectionner un élément dans le menu déroulant. Vous pouvez également cliquer avec le bouton droit sur le terminal de sortie et sélectionner l'élément dans le menu local de **Sélectionner un élément**.

Par exemple, si vous utilisez la fonction Désassembler avec le cluster de la figure 5-16, elle comporte quatre terminaux de sortie correspondant aux quatre commandes du cluster. Si vous utilisez la fonction Désassembler avec ce cluster, vous devez connaître l'ordre du cluster pour associer le terminal booléen correct du cluster désassemblé à son interrupteur. Dans cet exemple, les éléments sont classés de haut en bas à partir de l'élément 0. Si vous utilisez la fonction Désassembler par nom, vous pouvez avoir un nombre arbitraire de terminaux de sortie et accéder aux éléments individuels par leur nom quel que soit leur ordre dans le cluster.

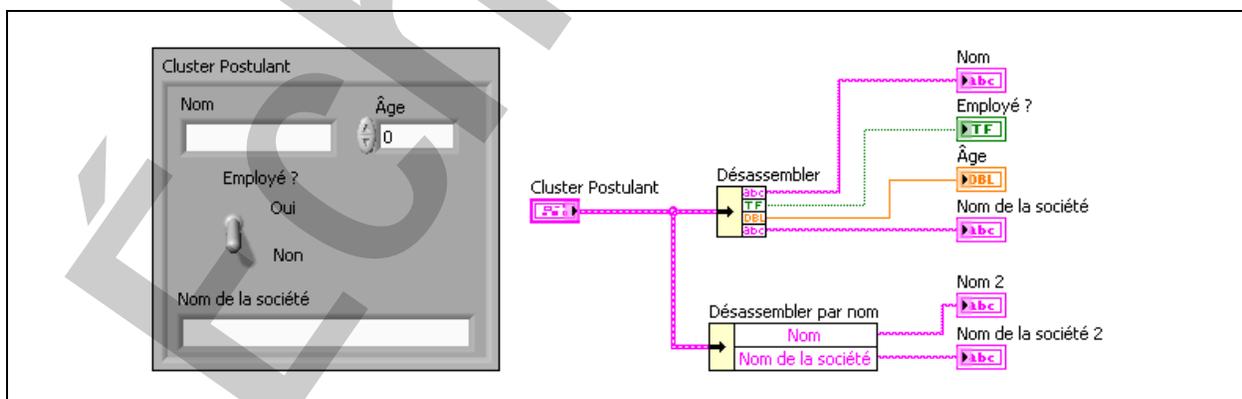


Figure 5-16. Désassembler et Désassembler par nom

Clusters d'erreur

LabVIEW comporte un cluster personnalisé appelé cluster d'erreur. LabVIEW utilise les clusters d'erreur pour transmettre des informations sur les erreurs. Un cluster d'erreur contient les éléments suivants :

- **état** représente une valeur booléenne qui renvoie la valeur VRAI si une erreur s'est produite.
- **code** représente un entier 32 bits signé qui identifie l'erreur numériquement. Un code d'erreur différent de zéro couplé avec un **état** FAUX signale une mise en garde plutôt qu'une erreur.
- **source** représente une chaîne qui identifie l'endroit où l'erreur s'est produite.

Pour obtenir plus d'informations concernant l'utilisation des clusters d'erreur, reportez-vous à la leçon 3, *Identification des problèmes et mise au point des VIs*, de ce manuel ainsi qu'à la rubrique *Gestion des erreurs de l'Aide LabVIEW*.

Pour mettre en pratique les concepts traités dans cette section, faites l'exercice 5-2.

C. Définitions de type

Utilisez des définitions de type pour définir des tableaux et des clusters personnalisés.

Commandes personnalisées

Utilisez les commandes et indicateurs personnalisés pour augmenter le nombre d'objets disponibles sur la face-avant. Dans une application, vous pouvez créer des objets d'interface utilisateur personnalisés dont l'apparence est différente des commandes et indicateurs standard de LabVIEW. Vous pouvez enregistrer une commande ou un indicateur personnalisé que vous avez créé dans un répertoire ou une LLB pour pouvoir l'utiliser sur d'autres faces-avant. Vous pouvez aussi créer une icône pour la commande ou l'indicateur personnalisé et l'ajouter à la palette

Commandes.

Reportez-vous à la rubrique *Création de commandes et d'indicateurs personnalisés, et de définitions de type* de l'*Aide LabVIEW* pour obtenir plus d'informations concernant la création et l'utilisation de commandes personnalisées et de définitions de type.

Utilisez la fenêtre de l'éditeur de commande pour personnaliser les commandes et les indicateurs. Par exemple, vous pouvez changer la taille,

la couleur et la position relative des éléments d'une commande ou d'un indicateur, et importer des images dans la commande ou l'indicateur.

Vous pouvez afficher la fenêtre de l'éditeur de commande de plusieurs façons :

- Cliquez avec le bouton droit sur la commande ou l'indicateur de la face-avant et sélectionnez **Avancé»Personnaliser** dans le menu local.
- Sélectionnez une commande ou un indicateur de la face-avant avec l'outil Flèche et sélectionnez **Édition»Personnaliser la commande**.
- Utilisez la boîte de dialogue **Nouveau**.

La fenêtre de l'éditeur de commande s'ouvre et affiche l'objet sélectionné. L'éditeur de commande a deux modes : le mode **Édition** et le mode **Personnalisé**.

La barre d'outils de la fenêtre de l'Éditeur de commande indique si vous êtes en mode **Édition** ou en mode **Personnalisé**. La fenêtre de l'éditeur de commande s'ouvre en mode **Édition**. Cliquez sur le bouton **Passer en mode Personnalisé** pour passer en mode personnalisé. Pour repasser en mode d'édition, cliquez sur le bouton **Passer en mode Édition**. Vous pouvez aussi passer d'un mode à l'autre en sélectionnant **Exécution»Passer en mode Personnalisé** ou **Exécution»Passer en mode Édition**.

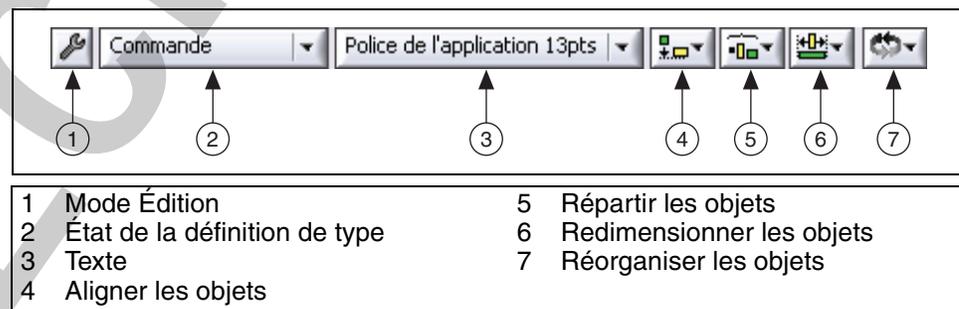


Utilisez le mode **Édition** pour changer la taille ou la couleur d'une commande ou d'un indicateur et pour sélectionner des options dans le menu local comme vous le feriez en mode **Édition** sur une face-avant.

Utilisez le mode **Personnalisé** pour changer considérablement les commandes et les indicateurs en modifiant leurs composantes individuelles.

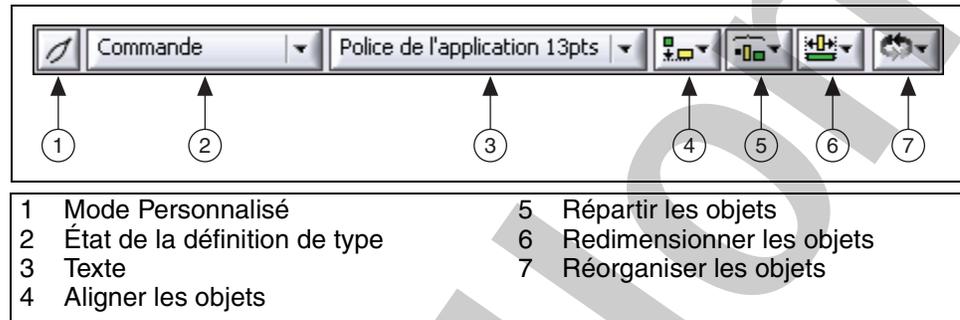
Mode Édition

En mode **Édition**, vous pouvez cliquer avec le bouton droit sur la commande et manipuler ses paramètres tout comme dans l'environnement de programmation LabVIEW.



Mode Personnalisé

En mode Personnalisé, vous pouvez déplacer les différentes composantes de la commande les unes par rapport aux autres. Pour obtenir une liste de ce que vous pouvez manipuler en mode Personnalisé, sélectionnez **Fenêtre» Visualiser la fenêtre des composants**.



Vous pouvez personnaliser une commande en changeant l'état de sa définition de type. Vous pouvez enregistrer une commande en tant que commande, définition de type ou définition de type stricte, en fonction de la sélection visible dans le menu déroulant d'**état de la définition de type**. L'option commande est identique à une commande sélectionnée sur la palette **Commandes**. Vous pouvez la modifier selon vos besoins et chaque copie, modifiée ou non, conserve ses propriétés individuelles.

Enregistrement de commandes personnalisées

Une fois une commande personnalisée créée, vous pouvez l'enregistrer pour un usage ultérieur. Par défaut, les commandes enregistrées sur disque ont l'extension `.ctl`.

Vous pouvez aussi utiliser l'éditeur de commande pour enregistrer les commandes avec vos propres paramètres par défaut. Par exemple, vous pouvez utiliser l'éditeur de commande pour modifier les valeurs par défaut d'un graphe, l'enregistrer et l'utiliser plus tard dans d'autres VIs.

Définitions de type

Utilisez les définitions de type et les définitions de type strictes pour lier toutes les instances d'une commande ou d'un indicateur personnalisé à un fichier de commande ou d'indicateur personnalisé que vous avez enregistré. Vous pouvez éditer toutes les instances de la commande ou de l'indicateur personnalisé en modifiant uniquement le fichier de commande ou d'indicateur personnalisé, ce qui est utile si vous utilisez la même commande ou le même indicateur dans plusieurs VIs.

Lorsque vous placez une commande ou un indicateur personnalisé dans un VI, il n'existe aucune connexion entre la commande ou l'indicateur personnalisé que vous avez enregistré et l'instance de cette commande ou de

cet indicateur dans le VI. Chaque instance de commande ou d'indicateur personnalisé est une copie indépendante. Par conséquent, les modifications que vous effectuez sur une commande ou un indicateur personnalisé n'affectent pas les VIs qui l'utilisent déjà. Si vous voulez établir un lien entre les instances d'une commande ou d'un indicateur personnalisé et le fichier qui lui correspond, enregistrez la commande ou l'indicateur personnalisé en tant que définition de type ou définition de type stricte. Toutes les instances d'une définition de type ou d'une définition de type stricte sont liées au fichier d'origine à partir duquel elles ont été créées.

Lorsque vous enregistrez une commande ou un indicateur personnalisé comme définition de type ou définition de type stricte, les modifications de type de données de la définition de type ou la définition de type stricte affectent toutes les instances de la définition de type ou la définition de type stricte dans tous les VIs qui l'utilisent. De plus, les modifications esthétiques que vous apportez à une définition de type stricte affectent toutes les instances de cette définition de type stricte sur la face-avant.

Les définitions de type identifient le type correct des données pour chaque instance d'une commande ou d'un indicateur personnalisé. Lorsque le type de données d'une définition de type change, toutes les instances de cette définition de type sont automatiquement mises à jour. En d'autres termes, le type de données des instances de la définition de type change dans tous les VIs où elle est utilisée. Cependant, comme les définitions de type n'identifient que le type de données, seules les valeurs qui font partie du type de données sont mises à jour. Par exemple, sur les commandes numériques, la gamme de données ne fait pas partie du type de données. Par conséquent, les définitions de type des commandes numériques ne définissent pas la gamme de données des instances des définitions de type. De plus, comme les noms d'éléments des commandes de type menu déroulant ne définissent pas le type des données, le changement de ces noms dans une définition de type ne changera pas les noms d'éléments dans les instances de la définition de type. Néanmoins, si vous changez les noms d'éléments dans la définition de type pour une commande de type énuméré, les instances sont mises à jour car les noms d'éléments font partie du type de données. Une instance d'une définition de type peut comporter un sous-titre, une étiquette, une description, une info-bulle, une taille par défaut, une couleur ou un style de commande ou d'indicateur qui lui sont propres, comme un bouton rotatif plutôt qu'une glissière.

Si vous changez le type de données dans une définition de type, LabVIEW remplace l'ancienne valeur par défaut par le nouveau type de données dans les instances de la définition de type si c'est possible. LabVIEW ne peut pas conserver la valeur par défaut des instances si le type de données est transformé en type incompatible, comme de commande ou indicateur numérique à commande ou indicateur chaîne. Si le type de données d'une

définition de type est transformé en type de données incompatible avec la définition de type précédente, LabVIEW définit la valeur par défaut des instances sur la valeur par défaut que vous avez spécifiée dans le fichier `.ctl`. Si vous ne spécifiez pas de valeur par défaut, LabVIEW utilise la valeur par défaut pour ce type de données. Par exemple, si vous changez la définition de type d'un type numérique à un type chaîne, LabVIEW remplace les valeurs par défaut associées à l'ancien type de données numérique par des chaînes vides.

Définitions de type strictes

Une définition de type stricte force tout ce qui concerne une instance à être identique à la définition de type stricte, à l'exception du sous-titre, de l'étiquette, de la description, de l'info-bulle et de la valeur par défaut. Comme pour les définitions de type, le type de données d'une définition de type stricte reste le même où que vous utilisiez la définition de type stricte. Les définitions de type strictes définissent aussi d'autres valeurs, comme la vérification de la gamme sur les commandes numériques et les noms d'éléments dans les commandes de type menu déroulant. Les seules propriétés de VI Serveur disponibles pour les définitions de type strictes sont celles qui affectent l'apparence de la commande ou de l'indicateur, comme Visible, Désactivé, Focus Clavier, Clignotant, Position et Limites.

Vous ne pouvez pas empêcher une instance d'une définition de type stricte d'être mise à jour automatiquement à moins de supprimer le lien entre l'instance et la définition de type stricte.

Les définitions de type et les définitions de type strictes servent à créer une commande personnalisée en utilisant un cluster constitué de nombreuses commandes. Si vous avez besoin d'ajouter une nouvelle commande et de faire passer une nouvelle valeur à tous les sous-VIs, vous pouvez ajouter la nouvelle commande au cluster de commandes personnalisées. Cela évite ainsi d'avoir à ajouter la nouvelle commande à la face-avant de chaque sous-VI et d'avoir à créer de nouveaux fils de liaison et terminaux.

Pour mettre en pratique les concepts traités dans cette section, faites l'exercice 5-3.

Auto-évaluation : quiz

1. Vous pouvez créer un tableau de tableaux.
 - a. Vrai
 - b. Faux
2. Vous avez deux tableaux en entrée câblés à une boucle For. L'auto-indexation est activée sur les deux tunnels. Un tableau a 10 éléments, l'autre en a cinq. La valeur 7 est câblée au terminal de décompte, comme l'illustre la figure 5-17. Quelle est la valeur de l'indicateur Itérations à la fin de l'exécution de ce VI ?

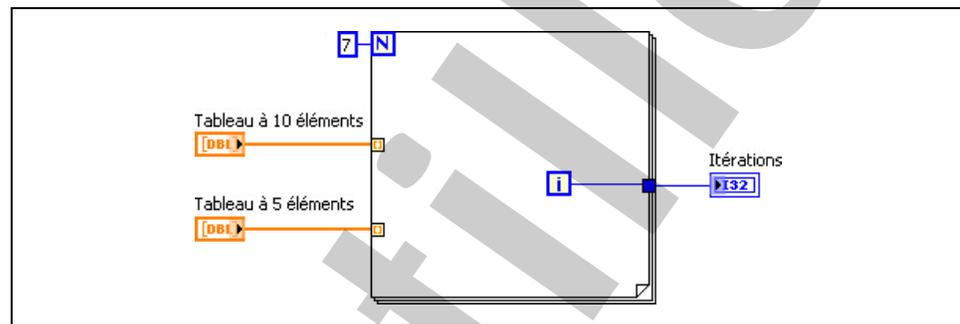


Figure 5-17. Quelle est la valeur de l'indicateur Itération ?

3. Vous personnalisez une commande, sélectionnez **Commande** dans le menu déroulant de **Type de commande** et enregistrez la commande sous forme de fichier .ctl. Vous utilisez ensuite une instance de la commande personnalisée sur votre fenêtre de face-avant. Si vous ouvrez le fichier .ctl et que vous modifiez la commande, la commande de la face-avant change-t-elle ?
 - a. Oui
 - b. Non
4. Vous entrez des données qui représentent un cercle. Les données représentant un cercle comprennent trois nombres double précision : position X, position Y et rayon. Il vous faudra peut-être à l'avenir développer toutes les instances des données de cercle pour inclure la couleur du cercle, représentée sous forme d'entier. Comment devriez-vous représenter le cercle sur votre face-avant ?
 - a. Trois commandes différentes pour les deux positions et le rayon.
 - b. Un cluster qui contient toutes les données.
 - c. Une commande personnalisée qui contient un cluster.
 - d. Une définition de type qui contient un cluster.
 - e. Un tableau à trois éléments.

Échantillon

Auto-évaluation : réponses du quiz

1. Vous pouvez créer un tableau de tableaux.
 - a. Vrai
 - b. Faux**

Vous ne pouvez pas faire glisser un type de données tableau dans un tableau vide. Par contre, vous pouvez créer des tableaux à deux dimensions.

2. Vous avez deux tableaux en entrée câblés à une boucle For. L'auto-indexation est activée sur les deux tunnels. Un tableau a 10 éléments, l'autre en a cinq. La valeur 7 est câblée au terminal de décompte de la boucle, comme l'illustre la figure suivante. Quelle est la valeur de l'indicateur Itérations à la fin de l'exécution de ce VI ?

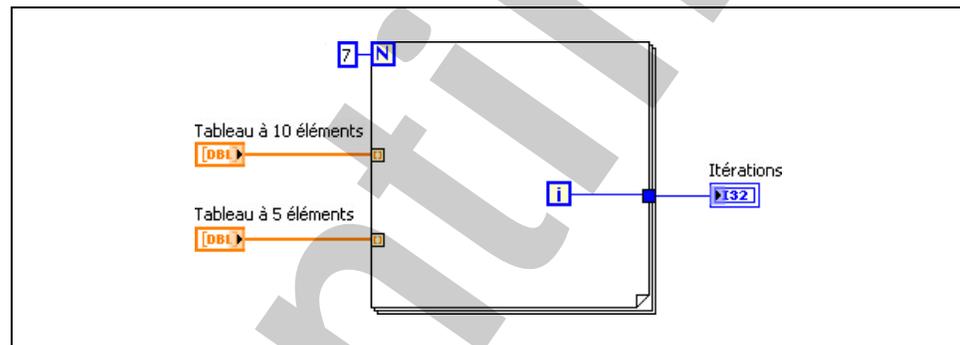


Figure 5-18. Quelle est la valeur de l'indicateur Itération ?

Valeur d'Itérations = 4

LabVIEW ne dépasse pas la taille du plus petit tableau. Ceci évite les erreurs de programmation. Les fonctions mathématiques de LabVIEW se comportent de la même façon : si vous câblez un tableau à 10 éléments à l'entrée x de la fonction Ajouter et un tableau à 5 éléments à l'entrée y de cette fonction, la sortie est un tableau à 5 éléments.

Bien que la boucle For s'exécute 5 fois, la valeur de l'indicateur Itérations est 4 car les itérations commencent à 0.

3. Vous personnalisez une commande, sélectionnez **Commande** dans le menu déroulant de **Type de commande** et enregistrez la commande sous forme de fichier `.ctl`. Vous utilisez ensuite une instance de la commande personnalisée sur votre fenêtre de face-avant. Si vous ouvrez le fichier `.ctl` et que vous modifiez la commande, la commande de la face-avant change-t-elle ?
 - a. Oui
 - b. Non**

4. Vous entrez des données qui représentent un cercle. Les données représentant un cercle comprennent trois nombres double précision : position X, position Y et rayon. Il vous faudra peut-être à l'avenir développer toutes les instances des données de cercle pour inclure la couleur du cercle, représentée sous forme d'entier. Comment devriez-vous représenter le cercle sur votre face-avant ?
 - a. Trois commandes différentes pour les deux positions et le rayon.
 - b. Un cluster qui contient toutes les données.
 - c. Une commande personnalisée qui contient un cluster.
 - d. Une définition de type qui contient un cluster.**
 - e. Un tableau à trois éléments.

Notes

Échantillon

Notes

Échantillon