

LabVIEW™ Core 1

Manuale del Corso

Versione del software del corso 2011

Edizione Agosto 2011

Codice 325290C-0115

Copyright

© 1993–2011 National Instruments Corporation. Tutti i diritti riservati.

In base alle norme sul copyright, questa pubblicazione non può essere riprodotta o trasmessa in alcuna forma, elettronica o meccanica, incluso fotocopie, registrazione, conservazione in un sistema informativo, o traduzione completa o parziale, senza il preventivo consenso scritto di National Instruments Corporation.

National Instruments rispetta la proprietà intellettuale altrui, e richiediamo ai nostri utenti di fare altrettanto. Il software NI è protetto da copyright e dalle leggi in materia di proprietà intellettuale. Qualora il software NI sia usato per riprodurre software o altro materiale appartenente a terze parti, potrete usare il software NI solo per riprodurre materiali che potete utilizzare secondo i termini e condizioni delle licenze applicabili o di altre previsioni legali.

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

Marchi

CVI, LabVIEW, National Instruments, NI, ni.com, il logo della società National Instruments ed il logo raffigurante l'Aquila costituiscono marchi di National Instruments Corporation. Si prega di rivolgersi a *Trademark Information* presso ni.com/trademarks in relazione agli altri marchi di 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. Altri nomi di prodotti o di aziende qui menzionati sono marchi o nomi commerciali di proprietà delle rispettive aziende.

I membri del National Instruments Alliance Partner Program sono entità commerciali indipendenti da National Instruments e non hanno relazioni di agenzia, partnership, joint-venture o associazione con National Instruments.

Brevetti

Per i brevetti che coprono i prodotti/la tecnologia National Instruments, si prega di fare riferimento alla fonte appropriata: a

Help»Patents nel vostro software, al file `patents.txt` nei vostri documenti, o a *National Instruments Patent Notice* su ni.com/patents.

Supporto tecnico internazionale e informazione di prodotto

ni.com

Sedi internazionali

Per accedere ai siti Web degli uffici locali visitate ni.com/niglobal, troverete informazioni aggiornate su contatti, numeri telefonici del supporto tecnico, indirizzi email ed eventi.

Sede centrale di National Instruments Corporation

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

Per ulteriori informazioni di supporto, fate riferimento all'appendice *Informazioni e risorse aggiuntive*. Per commenti sulla documentazione di National Instruments, fate riferimento al sito web di National Instruments ni.com/info e inserite il codice informativo feedback.

Contenuti

Guida per lo studente

A. Certificazione NI.....	vii
B. Descrizione del corso.....	viii
C. Prerequisiti.....	ix
D. Installazione del software per il corso.....	x
E. Obiettivi del corso.....	x
F. Convenzioni adottate nel corso.....	xi

Lezione 1

Impostazione Hardware

A. Hardware DAQ.....	1-2
B. Uso del software DAQ.....	1-9
C. Controllo della strumentazione.....	1-12
D. GPIB.....	1-12
E. Porta di comunicazione seriale.....	1-14
F. Uso del software di controllo della strumentazione.....	1-17
G. Progetto del corso.....	1-19

Lezione 2

Orientarsi in LabVIEW

A. Virtual Instruments (VI).....	2-2
B. Componenti di un VI.....	2-2
C. Aprire un VI.....	2-4
D. Finestra Project Explorer.....	2-8
E. Pannello frontale.....	2-14
F. Diagramma a blocchi.....	2-22
G. Ricerca di Controlli, VI e Funzioni.....	2-31
H. Scelta di uno strumento.....	2-34
I. Flusso dei dati.....	2-41
J. Realizzazione di un semplice VI.....	2-42

Lezione 3

Troubleshooting e debug dei VI

A. Strumenti di Help in LabVIEW.....	3-2
B. Correzione di VI non eseguibili.....	3-5
C. Tecniche di debug.....	3-6
D. Dati indefiniti o inattesi.....	3-14
E. Verifica e gestione degli errori.....	3-15

Lezione 4

Implementazione di un VI

A. Progettazione del pannello frontale	4-2
B. Tipi di dati in LabVIEW	4-9
C. Documentazione del codice	4-19
D. While Loop	4-22
E. For Loop	4-26
F. Temporizzazione di un VI	4-30
G. Trasferimento iterativo dei dati.....	4-31
H. Rappresentazione dei dati	4-34
I. Strutture Case.....	4-40

Lezione 5

Aggregazione dei dati

A. Array	5-2
B. Cluster	5-7
C. Definizioni di tipo.....	5-14

Lezione 6

Gestione delle risorse

A. Apprendere il file I/O.....	6-2
B. Apprendere il file I/O High-Level	6-3
C. Apprendere il file I/O Low-Level.....	6-4
D. Programmazione DAQ	6-6
E. Programmazione per il controllo della strumentazione	6-8
F. Driver degli strumenti.....	6-11

Lezione 7

Sviluppo di applicazioni modulari

A. Capire la modularità.....	7-2
B. Costruzione dell'icona e del connector pane	7-3
C. Uso dei subVI	7-9

Lezione 8

Tecniche di progettazione e design pattern comuni

A. Programmazione sequenziale	8-2
B. Programmazione a stati.....	8-4
C. Macchine a stati	8-5
D. Parallelismo	8-14

Lezione 9

Uso delle variabili

A. Parallelismo	9-2
B. Variabili	9-4
C. Variabili Functional Global	9-14
D. Race Condition	9-17

Appendice A

Fondamenti della misura

A. Sistemi di misura basati su computer	A-2
B. Concetti della misura	A-3
C. Come aumentare la qualità della misura.....	A-13

Appendice B

Informazioni e risorse aggiuntive

Glossario

Indice

Esempio

Aggregazione dei dati

Talvolta è vantaggioso raggruppare dati correlati gli uni agli altri. In LabVIEW si usano array e cluster per aggregare dati correlati. Gli array combinano dati dello stesso tipo in un'unica struttura dati, mentre i cluster combinano dati di tipo diverso in un'unica struttura dati. Per definire array e cluster personalizzati si usano definizioni di tipo. Questa lezione spiega gli array, i cluster, le definizioni di tipo e le applicazioni in cui il loro utilizzo può essere vantaggioso.

Argomenti

- A. Array
- B. Cluster
- C. Definizioni di tipo

A. Array

Un array è caratterizzato da elementi e dimensioni. Gli elementi sono i dati che costituiscono l'array. Una dimensione è la lunghezza, l'altezza o profondità dell'array. Un array può avere una o più dimensioni ed un numero di elementi per dimensione pari al massimo a $(2^{31}) - 1$, se la memoria lo permette.

Potete costruire array di numeri, di booleani, di percorsi, di stringhe, di forme d'onda e di cluster. Prendete in considerazione l'uso di array quando lavorate con un insieme di dati simili e quando eseguite calcoli ripetitivi. Gli array sono ideali per la memorizzazione di dati che raccogliete da forme d'onda o di dati generati nei cicli, in cui ogni iterazione di un ciclo produce un elemento dell'array.



Nota In LabVIEW gli indici di array partono da zero. L'indice del primo elemento nell'array, a prescindere dalle sue dimensioni, è zero

Restrizioni

Non potete creare array di array. Tuttavia potete utilizzare un array multidimensionale o creare un array di cluster in cui ogni cluster contiene uno o più array. Inoltre non potete creare un array di controlli subpanel, tab, .NET, ActiveX, chart o XY graph multitraccia. Fate riferimento alla sezione cluster di questa lezione per avere maggiori informazioni sui cluster.

Un esempio di un semplice array è un array di testo che elenca i dodici mesi dell'anno. LabVIEW lo rappresenta come un array 1D di stringhe con dodici elementi.

Gli elementi dell'array sono ordinati. Un array utilizza un indice affinché possiate prontamente accedere ad ogni elemento particolare. L'indice è in base zero, cioè parte da 0 e arriva a $n - 1$, in cui n è il numero di elementi presenti nell'array. Per esempio, con $n = 12$ per i dodici mesi dell'anno, l'indice varia da 0 a 11. Marzo è il terzo mese, con un indice pari a 2.

La Figura 5-1 mostra un esempio di un array di numeri. Il primo elemento mostrato nell'array (3.00) ha l'indice 1 e il secondo elemento (1.00) ha l'indice 2. L'elemento selezionato nell'index display fa sempre riferimento all'elemento mostrato nell'angolo superiore sinistro dell'element display.

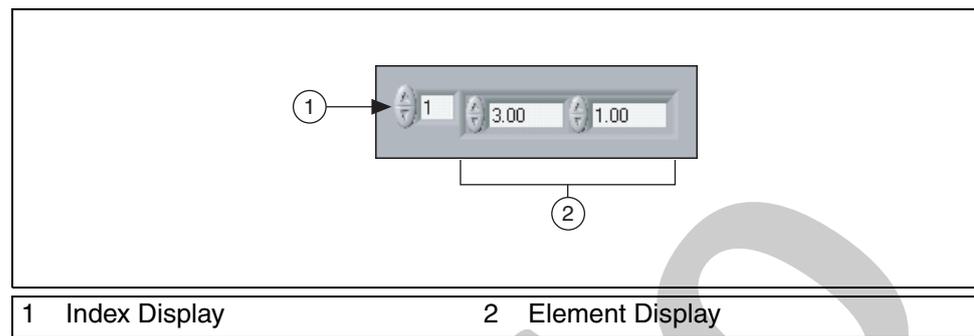


Figura 5-1. Controllo array di numeri

Creazione di array di controlli e indicatori

Create un controllo o indicatore ad array sul pannello frontale inserendo un contenitore array sul pannello frontale, come mostrato nel seguente pannello frontale, e trascinando nell'array un oggetto o elemento, che può essere un numero, un valore booleano, una stringa, un path, un refnum oppure un indicatore o un controllo cluster.

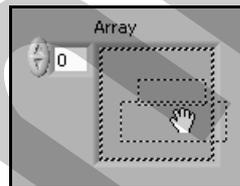


Figura 5-2. Inserimento di un controllo numerico in un contenitore array

Se tentate di trascinare nel contenitore array un controllo o un indicatore non valido non siete in grado di inserire nel contenitore array il controllo o l'indicatore.

Dovete inserire un oggetto nel contenitore array prima che possiate utilizzare l'array nel diagramma a blocchi. Altrimenti, il terminale dell'array appare nero con una parentesi vuota e non ha alcun tipo di dato associato ad esso.

Array bidimensionali

Gli esempi precedenti utilizzano array 1D. Un array 2D memorizza elementi in una griglia. Esso richiede un indice di colonna e un indice di riga per individuare un elemento, entrambi partono da zero. La Figura 5-3 mostra un array 2D di 8 colonne per 8 righe, che contiene $8 \times 8 = 64$ elementi.

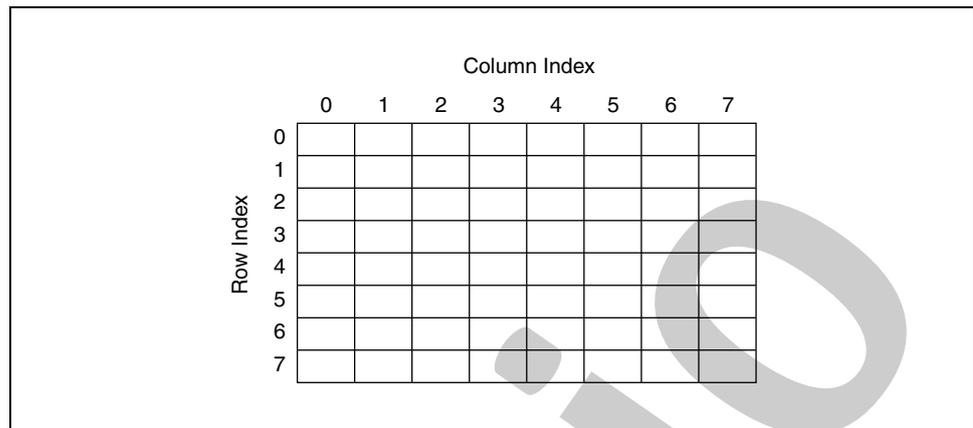


Figura 5-3. Array 2D

Per aggiungere un array multidimensionale al pannello frontale, cliccate con il tasto destro sull'index display e selezionate **Add Dimension** dal menu rapido. Potete anche ridimensionare l'index display finché non ottenete tante dimensioni quante ne desiderate.

Inizializzazione di array

Potete inizializzare un array o lasciarlo non inizializzato. Quando un array è inizializzato, definite il numero di elementi in ogni dimensione e il contenuto di ogni elemento. Un array non inizializzato ha un numero fisso di dimensioni ma nessun elemento. La Figura 5-4 mostra un controllo array 2D non inizializzato. Osservate che gli elementi sono tutti opachi. Questo indica che l'array non è inizializzato.



Figura 5-4. Array 2D non inizializzato

Nella Figura 5-5, sono inizializzati sei elementi.

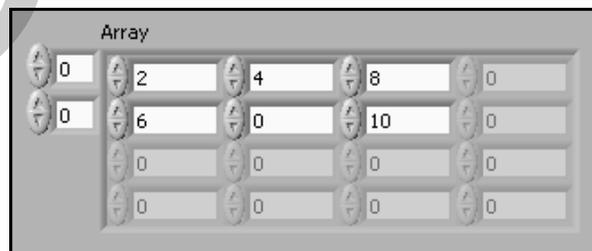


Figura 5-5. Array 2D inizializzato con sei elementi

In un array 2D, una volta che avete inizializzato un elemento in una riga, gli elementi rimanenti in quella riga sono inizializzati e popolati con valori di default per quel tipo di dato. Per esempio, nella Figura 5-6, se voi digitate 4 nell'elemento della prima colonna, terza riga, gli elementi nella seconda e terza colonna nella terza riga vengono automaticamente popolati con 0.

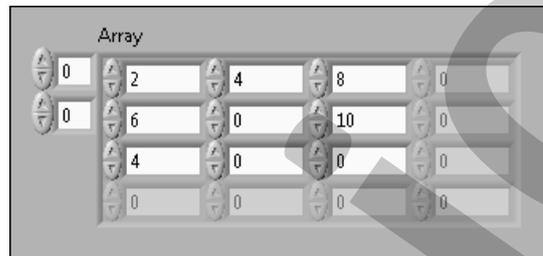


Figura 5-6. Array popolato automaticamente con zero

Creazione di costanti array

Per creare una costante array nel diagramma a blocchi, selezionate una costante array nella palette **Functions**, ponete il contenitore array nel diagramma a blocchi ed inserite una costante stringa, una costante numerica, una costante booleana o una costante cluster nel contenitore array. Potete utilizzare array di costanti per memorizzare dati costanti o come base per un confronto con un altro array.

Auto-index di ingressi array

Se collegate un array verso o da un For Loop o un While Loop, potete collegare ciascuna iterazione del ciclo ad un elemento dell'array abilitando l'auto-index. L'immagine del tunnel si trasforma da un quadrato solido nell'immagine ad indicare l'auto-index. Cliccate con il tasto destro del mouse sul tunnel e selezionate **Enable Indexing** o **Disable Indexing** dal menu rapido per commutare lo stato del tunnel.



Ingressi array

Se abilitate l'auto-index in un array collegato ad un terminale d'ingresso del For Loop, LabVIEW imposta il count terminal alla dimensione dell'array e quindi non dovete collegare il count terminal. Siccome potete utilizzare i For Loop per processare array un elemento alla volta, LabVIEW abilita l'auto-index di default per ogni array che collegate ad un For Loop. Potete disabilitare l'auto-index se non avete bisogno di processare array un elemento alla volta.

Nella Figura 5-7, il For Loop viene eseguito un numero di volte pari al numero di elementi nell'array. Normalmente, se il count terminal del For Loop non è collegato, la freccia di run è spezzata. Tuttavia, in questo caso, la freccia di run non è spezzata.

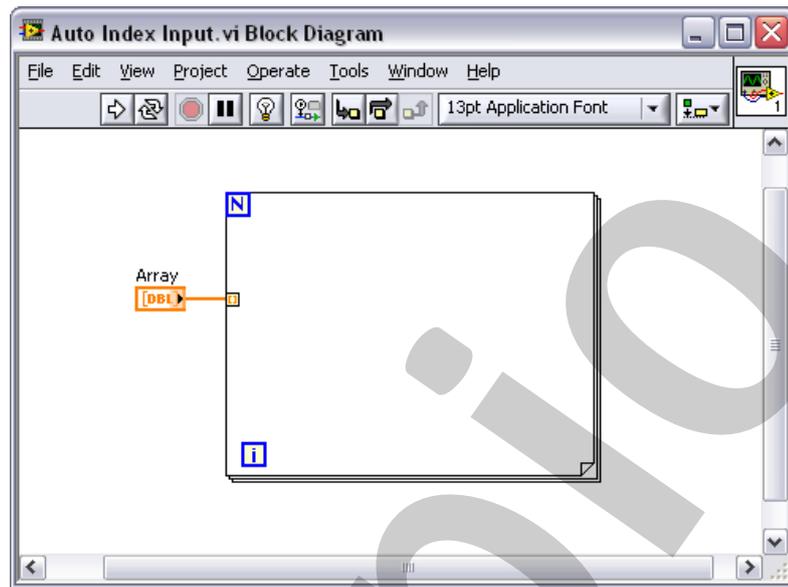


Figura 5-7. Array utilizzato per impostare il conteggio del For Loop

Se abilitate l'auto-index per più di un tunnel o se collegate il count terminal, il numero di iterazioni diventa pari al valore possibile inferiore. Per esempio, se due array con auto-index entrano nel ciclo, con 10 e 20 elementi rispettivamente, e voi collegate un valore di 15 al count terminal, il ciclo viene comunque eseguito ancora 10 volte, indicizzando tutti gli elementi del primo array, ma solo i primi 10 elementi del secondo array.

Uscite array

Quando effettuate l'auto-index di un tunnel di uscita array, l'array di uscita riceve un nuovo elemento ad ogni iterazione del ciclo. Quindi, gli array di uscita con auto-index hanno sempre dimensione uguale al numero di iterazioni.

Il collegamento dal tunnel di uscita all'indicatore array diventa più spesso nel momento in cui cambia in un array sulla cornice del ciclo e il tunnel di uscita contiene parentesi quadre a rappresentare un array, come mostrato in Figura 5-8.

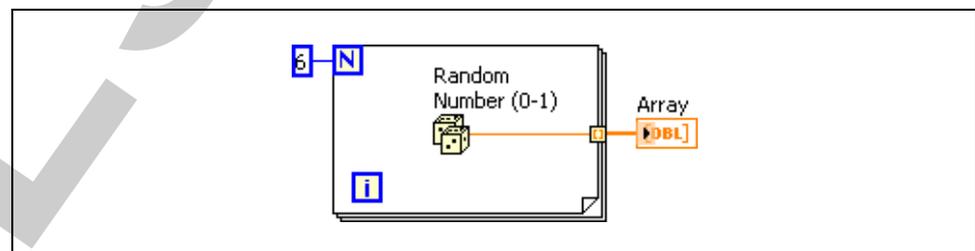


Figura 5-8. Uscita con auto-index

Cliccate con il tasto destro del mouse il tunnel sulla cornice del ciclo e selezionate **Enable Indexing** o **Disable Indexing** dal menu rapido per abilitare o disabilitare l'auto-index. Nei While Loop l'auto-index è disabilitato di default.

Per esempio, disabilitate l'auto-index se avete bisogno di passare solo l'ultimo valore in uscita dal tunnel.

Creazione di array bidimensionali

Potete utilizzare due For Loop, uno dentro l'altro, per creare un array 2D. Il For Loop esterno crea gli elementi di riga e quello interno gli elementi di colonna, come mostrato in Figura 5-9.

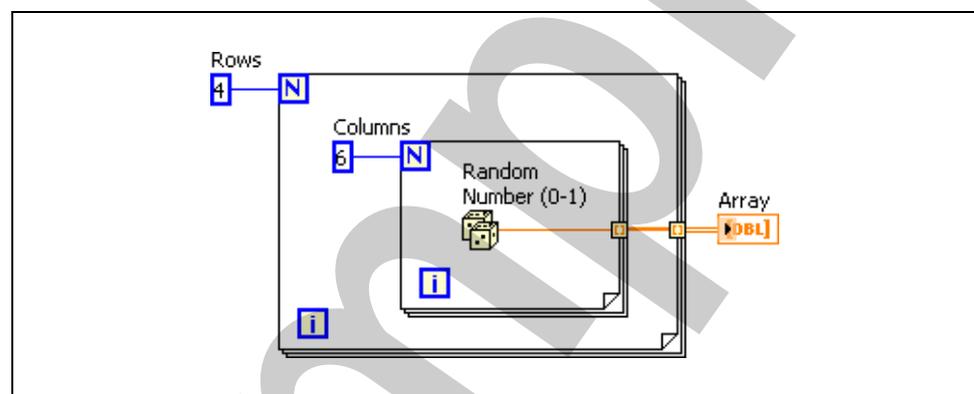


Figura 5-9. Creazione di un array 2D

Per fare pratica con i concetti di questa sezione, completate l'Esercizio 5-1.

B. Cluster

I cluster raggruppano elementi di tipo di dato diverso. Un esempio di cluster è il cluster degli errori di LabVIEW, che combina un valore booleano, un valore numerico ed una stringa. Un cluster è simile ad un record o ad una struct nei linguaggi di programmazione testuali.

Raggruppare diversi elementi nei cluster elimina collegamenti disordinati nel diagramma a blocchi e riduce il numero di terminali del connector pane di cui hanno bisogno i subVI. Il connector pane ha, al più, 28 terminali. Se il vostro pannello frontale contiene più di 28 controlli ed indicatori che volete passare ad un altro VI, raggruppate alcuni di essi in un cluster ed assegnate il cluster ad un terminale sul connector pane.

La maggior parte dei cluster nel diagramma a blocchi possiede collegamento e terminale associati rosa. I cluster di errore hanno collegamento e terminale associati giallo scuro. I cluster di valori numerici,

a cui talvolta ci si riferisce come punti, hanno collegamento e terminale associati marrone. Potete collegare cluster numerici marroni a funzioni Numeric, come Add o Square Root, per effettuare la stessa operazione contemporaneamente su tutti gli elementi del cluster.

Ordine degli elementi del cluster

Sebbene gli elementi di cluster e di array siano entrambi ordinati, dovete separare tutti gli elementi del cluster usando la funzione Unbundle. Potete usare la funzione Unbundle By Name per separare elementi di cluster in base al nome. Se usate la funzione Unbundle by Name, ogni elemento del cluster deve avere un'etichetta. I cluster differiscono inoltre dagli array avendo una dimensione fissa. Come un array, un cluster può essere un controllo o un indicatore. Un cluster non può contenere sia controlli che indicatori.

Creazione di controlli e indicatori cluster

Create un controllo o indicatore cluster sul pannello frontale inserendo un contenitore cluster sul pannello frontale, come mostrato nel seguente pannello frontale, e trascinando nel contenitore cluster un oggetto o elemento, che può essere un numero, un valore booleano, una stringa, un path, un refnum o un indicatore o controllo cluster.

Ridimensionate il contenitore cluster trascinando il cursore mentre posizionate il contenitore cluster .

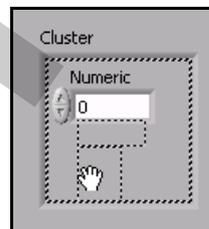


Figura 5-10. Creazione di un controllo cluster

La Figura 5-11 è un esempio di cluster contenente tre controlli: una stringa, uno switch booleano ed un numero. Un cluster è un controllo o un indicatore; esso non può contenere sia controlli che indicatori.

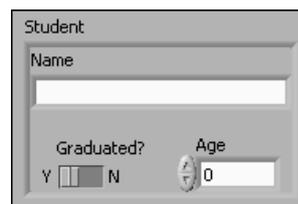


Figura 5-11. Esempio di controllo cluster

Creazione di cluster di costanti

Per creare un cluster di costanti nel diagramma a blocchi, selezionate un cluster di costanti nella palette **Functions**, ponete il contenitore del cluster nel diagramma a blocchi ed inserite una costante stringa, una costante numerica, una costante booleana o una costante cluster all'interno del contenitore del cluster. Potete utilizzare un cluster di costanti per memorizzare dati costanti o come base per un confronto con un altro cluster.

Se avete un controllo o indicatore cluster nel pannello frontale e volete creare un cluster di costanti contenente gli stessi elementi nel diagramma a blocchi, potete sia trascinare quel cluster dal pannello frontale al diagramma a blocchi sia cliccare con il tasto destro del mouse sul cluster sul diagramma a blocchi e selezionare **Create»Constant** dal menu rapido.

Ordine del cluster

Gli elementi del cluster possiedono un ordine logico scorrelato dalla loro posizione nella finestra contenitore. Il primo oggetto che inserite nel cluster è l'elemento 0, il secondo è l'elemento 1 e così via. Se cancellate un elemento, l'ordine si aggiusta automaticamente. L'ordine del cluster determina l'ordine in cui gli elementi appaiono come terminali nelle funzioni Bundle e Unbundle nel diagramma a blocchi. Potete visualizzare e modificare l'ordine del cluster cliccando con il tasto destro del mouse sulla cornice del cluster e selezionando **Reorder Controls In Cluster** dal menu rapido.

La barra degli strumenti e il cluster cambiano, come mostrato in Figura 5-12.

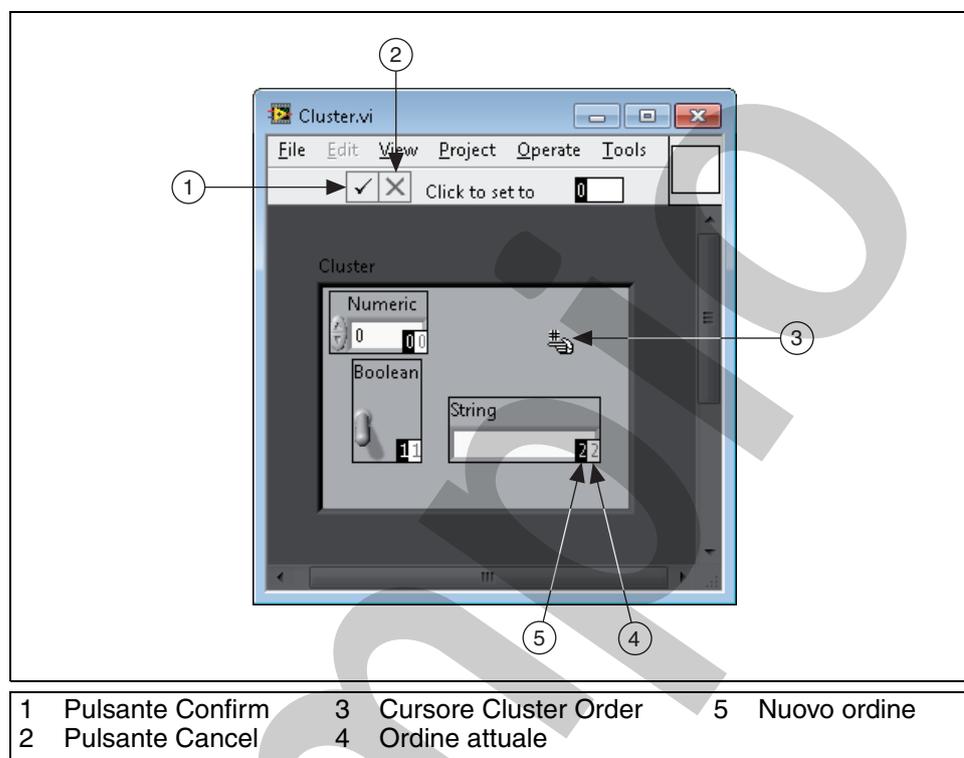


Figura 5-12. Riorganizzazione di un cluster

Il riquadro bianco su ogni elemento mostra la sua posizione attuale nell'ordine del cluster. Il riquadro nero mostra la nuova posizione dell'elemento. Per impostare l'ordine di un elemento del cluster, inserite il nuovo numero d'ordine nel riquadro di testo **Click to set to** e cliccate sull'elemento. L'ordine dell'elemento nel cluster cambia e l'ordine nel cluster degli altri elementi viene modificato di conseguenza. Salvate le modifiche cliccando sul pulsante **Confirm** della barra degli strumenti. Tornate all'ordine originale cliccando sul pulsante **Cancel**.

Uso di funzioni cluster

Utilizzate le funzioni Cluster per creare e manipolare cluster. Per esempio, potete svolgere task simili ai seguenti:

- Estrarre da un cluster singoli elementi di dati.
- Aggiungere ad un cluster elementi di dati individuali.
- Scomporre un cluster nei suoi elementi di dati individuali.

Utilizzate la funzione Bundle per assemblare un cluster, le funzioni Bundle e Bundle by Name per modificare un cluster e le funzioni Unbundle e Unbundle by Name per disassemblare cluster.

Potete anche inserire nel diagramma a blocchi le funzioni Bundle, Bundle by Name, Unbundle e Unbundle by Name cliccando con il tasto destro del mouse su un terminale del cluster sul diagramma a blocchi e selezionando **Cluster, Class & Variant Palette** dal menu rapido. Le funzioni Bundle e Unbundle contengono automaticamente il numero corretto di terminali. Le funzioni Bundle by Name e Unbundle by Name appaiono con il primo elemento nel cluster. Utilizzate lo strumento Positioning per ridimensionare le funzioni Bundle by Name e Unbundle by Name per mostrare altri elementi del cluster.

Assemblaggio di cluster

Utilizzate la funzione Bundle per assemblare un cluster a partire da elementi singoli o per modificare i valori di singoli elementi in un cluster esistente senza dover specificare nuovi valori per tutti gli elementi. Utilizzate lo strumento Positioning per ridimensionare la funzione o cliccate con il tasto destro del mouse su un ingresso di un elemento e selezionate **Add Input** dal menu rapido.

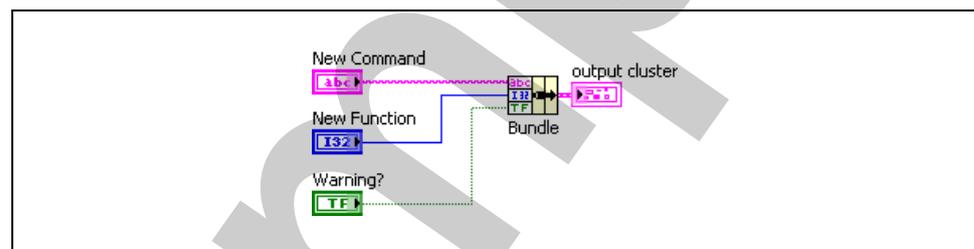


Figura 5-13. Assemblaggio di un cluster nel diagramma a blocchi

Modifica di un cluster

Se collegate l'ingresso del cluster, potete collegare solo gli elementi che volete modificare. Per esempio, l'Input Cluster mostrato in Figura 5-14 contiene tre controlli.

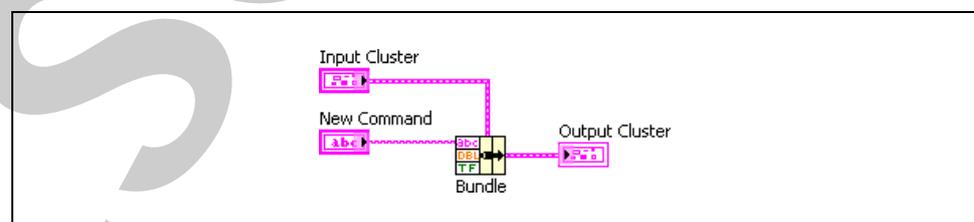


Figura 5-14. Bundle utilizzato per modificare un cluster

Se conoscete l'ordine del cluster, potete utilizzare la funzione Bundle per modificare il valore **Command** collegando gli elementi mostrati nella Figura 5-14.

Potete anche utilizzare la funzione Bundle by Name per sostituire o avere accesso ad elementi etichettati di un cluster esistente. La funzione Bundle by Name lavora analogamente alla funzione Bundle, ma invece di riferirsi ad elementi di cluster tramite l'ordine del cluster, si riferisce ad essi tramite le loro etichette. Potete accedere soltanto ad elementi dotati di propria etichetta. Il numero di ingressi non necessita di essere coincidente con il numero di elementi in **output cluster**.

Utilizzate lo strumento Operating per cliccare su un terminale d'ingresso e selezionare un elemento dal menu a tendina. Potete anche cliccare con il tasto destro del mouse sull'ingresso e selezionare l'elemento **Select Item** dal menu rapido.

Nella Figura 5-15, potete utilizzare la funzione Bundle by Name per modificare i valori di **Command** e **Function** con i valori di **New Command** e **New Function**.

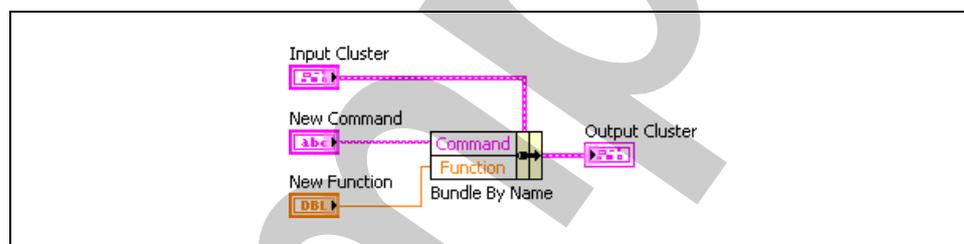


Figura 5-15. Bundle By Name utilizzato per modificare un cluster

Utilizzate la funzione Bundle by Name per strutture di dati che possono cambiare durante lo sviluppo. Se aggiungete un nuovo elemento al cluster o modificate il suo ordine, non avete bisogno di ricollegare la funzione Bundle by Name perché i nomi sono ancora validi.

Disassemblaggio di cluster

Utilizzate la funzione Unbundle per suddividere un cluster nei suoi singoli elementi.

Utilizzate la funzione Unbundle by Name per fornire gli elementi del cluster di cui specificate i nomi. Il numero di terminali di uscita non dipende dal numero di elementi del cluster d'ingresso.

Utilizzate lo strumento Operating per cliccare su un terminale d'uscita e selezionare un elemento dal menu a tendina. Potete anche cliccare con il tasto destro del mouse sull'ingresso e selezionare l'elemento **Select Item** dal menu rapido.

Per esempio, se utilizzate la funzione Unbundle con il cluster in Figura 5-16, esso ha quattro terminali di uscita che corrispondono ai quattro controlli nel cluster. Dovete conoscere l'ordine del cluster affinché possiate associare il

terminale booleano corretto del cluster disassemblato con lo switch corrispondente nel cluster. In questo esempio, gli elementi sono ordinati dall'alto verso il basso a partire dall'elemento 0. Se utilizzate la funzione Unbundle by Name, potete avere un numero arbitrario di terminali di uscita ed accedere ai singoli elementi per nome in qualsiasi ordine.

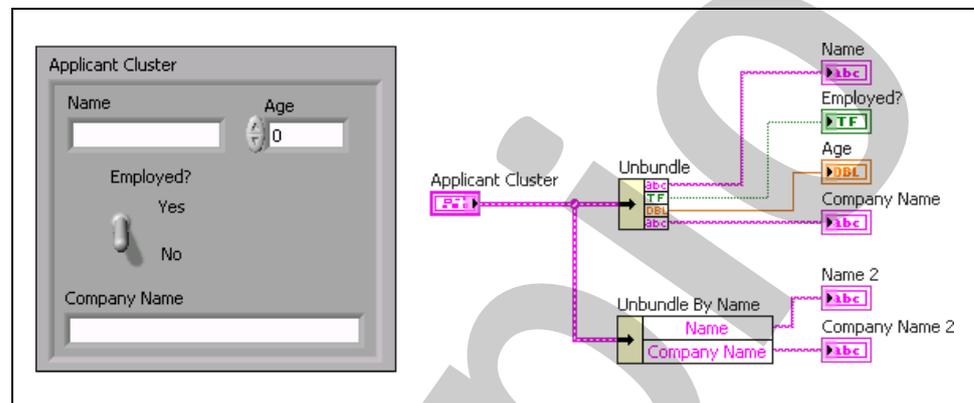


Figura 5-16. Unbundle e Unbundle by Name

Cluster di errore

LabVIEW contiene un cluster personalizzato chiamato cluster di errore. LabVIEW utilizza i cluster di errore per passare informazioni sugli errori. Un cluster di errore contiene gli elementi seguenti:

- **status** è un valore booleano che riporta TRUE in caso di errore.
- **code** è un numero intero con segno a 32 bit che identifica numericamente l'errore.
- **source** è una stringa che identifica dove è avvenuto l'errore.

Per maggiori informazioni sull'uso dei cluster di errore, fate riferimento alla Lezione 3, *Troubleshooting e debug dei VI*, di questo manuale e all'argomento *Handling Errors* del *LabVIEW Help*.

Per fare pratica con i concetti di questa sezione, completate l'Esercizio 5-2.

C. Definizioni di tipo

Potete usare le definizioni di tipo per definire array e cluster personalizzati.

Controlli personalizzati

Utilizzate controlli e indicatori personalizzati per estendere l'insieme disponibile di oggetti di un pannello frontale. Potete creare componenti personalizzati per l'interfaccia utente di un'applicazione con aspetto estetico diverso dai controlli e indicatori integrati di LabVIEW. Potete salvare in una directory o in una LLB un controllo o un indicatore personalizzato che avete creato ed utilizzare il controllo o l'indicatore su altri pannelli frontali. Potete anche creare un'icona per il controllo o l'indicatore personalizzato e aggiungerlo alla palette **Controls**.

Fate riferimento all'argomento *Creating Custom Controls, Indicators, and Type Definitions* del *LabVIEW Help* per ulteriori informazioni sulla creazione e l'utilizzo di controlli personalizzati e di definizioni di tipo.

Utilizzate la finestra Control Editor per personalizzare controlli e indicatori. Per esempio, potete cambiare le dimensioni, il colore e la posizione relativa degli elementi di un controllo o di un indicatore e importare immagini nel controllo o indicatore.

Potete visualizzare la finestra Control Editor nei seguenti modi:

- Cliccate con il tasto destro del mouse su un controllo o indicatore del pannello frontale e selezionate **Advanced»Customize** dal menu rapido.
- Utilizzate lo strumento Positioning per selezionare un controllo o indicatore del pannello frontale e selezionate **Edit»Customize Control**.
- Utilizzate la finestra di dialogo **New**.

Il Control Editor appare con l'oggetto del pannello frontale selezionato nella sua finestra. Il Control Editor ha due modalità, edit e customize.

La barra degli strumenti della finestra Control Editor indica se siete in modalità edit o customize. La finestra Control Editor si apre in modalità edit. Cliccate sul pulsante **Change to Customize Mode** per passare alla modalità customize. Per tornare alla modalità edit, cliccate sul pulsante **Change to Edit Mode**. Potete anche passare da una modalità all'altra selezionando **Operate»Change to Customize Mode** o **Operate»Change to Edit Mode**.

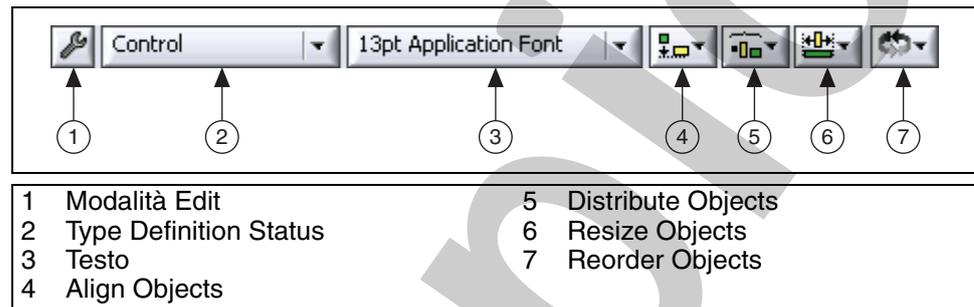
Utilizzate la modalità edit per cambiare le dimensioni o il colore di un controllo o di un indicatore e per selezionare le opzioni dal suo menu rapido, proprio come fate in modalità edit su un pannello frontale.



Utilizzate la modalità customize per apportare significativi cambiamenti ai controlli o indicatori cambiando le singole parti di un controllo o di un indicatore.

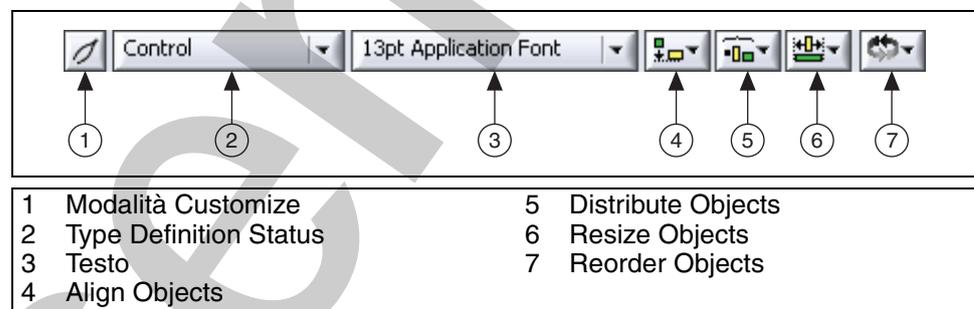
Modalità Edit

In modalità edit, potete cliccare con il tasto destro del mouse sul controllo e modificare le sue impostazioni come le vorreste nell'ambiente di programmazione di LabVIEW.



Modalità Customize

In modalità Customize, potete spostare i singoli componenti del controllo uno rispetto all'altro. Per un elenco completo di quello che potete manipolare in modalità customize, selezionate **Window»Show Parts Window**.



Un modo di personalizzare un controllo è di modificare il suo type definition status. Potete salvare un controllo come un control, una definizione di tipo (type def) o una definizione di tipo strict (strict type def), in funzione della selezione visibile nel controllo **Type Def. Status**. L'opzione control è la stessa di un controllo selezionato dalla palette **Controls**. Potete modificarlo come volete ed ogni copia che ne fate e modificate mantiene le sue proprietà.

Salvataggio dei controlli personalizzati

Dopo aver creato un controllo personalizzato, potete salvarlo per un uso successivo. Di default i controlli salvati su disco hanno un'estensione .ctl.

Potete anche utilizzare Control Editor per salvare controlli con le vostre impostazioni di default. Per esempio, potete utilizzare Control Editor per

modificare i valori di default di un waveform graph, salvarli e richiamarli successivamente in altri VI.

Definizioni di tipo

Utilizzate le definizioni di tipo e le definizioni di tipo strict per collegare tutte le istanze di un controllo o di un indicatore personalizzato ad un file salvato del controllo o indicatore personalizzato. Potete effettuare cambiamenti a tutte le istanze del controllo o indicatore personalizzato modificando solo il file salvato del controllo o indicatore personalizzato, il che è utile se utilizzate lo stesso controllo o indicatore personalizzato in diversi VI.

Quando inserite un controllo o indicatore personalizzato in un VI, non esiste collegamento tra il controllo o indicatore personalizzato che avete salvato e l'istanza del controllo o indicatore nel VI. Ogni istanza di un controllo o indicatore personalizzato è una copia indipendente e separata. Quindi i cambiamenti che apportate ad un file del controllo o indicatore personalizzato non influenzano i VI che già utilizzano quel controllo o indicatore personalizzato. Se volete collegare istanze di un controllo o indicatore personalizzato al file del controllo o indicatore personalizzato, salvate il controllo o indicatore personalizzato come definizione di tipo o come definizione di tipo strict. Tutte le istanze di definizione di tipo o di definizione di tipo strict si collegano al file originale da cui le avete create.

Quando salvate un controllo o indicatore personalizzato come definizione di tipo o definizione di tipo strict, ogni modifica che apportate alla definizione di tipo o alla definizione di tipo strict influenza tutte le istanze della definizione di tipo o della definizione di tipo strict in tutti i VI che le utilizzano. Inoltre cambiamenti cosmetici alla definizione di tipo strict influenzano tutte le istanze della definizione di tipo strict sul pannello frontale.

Le definizioni di tipo identificano il tipo di dato corretto per ogni istanza di un controllo o di un indicatore personalizzato. Quando il tipo di dato di una definizione di tipo cambia, tutte le istanze della definizione di tipo si aggiornano automaticamente. In altre parole, il tipo di dato delle istanze della definizione di tipo cambia in ogni VI in cui la definizione di tipo viene usata. Tuttavia, siccome le definizioni di tipo identificano solo il tipo di dato, solo i valori che fanno parte del tipo di dato si aggiornano. Per esempio, sui controlli numerici, il data range non fa parte del tipo di dato. Quindi le definizioni di tipo dei controlli numerici non definiscono il data range per le istanze delle definizioni di tipo. Inoltre, siccome i nomi delle voci nei controlli ring non definiscono il tipo di dato, cambiamenti ai nomi delle voci del controllo ring in una definizione di tipo non cambiano i nomi delle voci nelle istanze della definizione di tipo. Tuttavia, se modificate i nomi delle voci nella definizione di tipo per un controllo di tipo enumerated, le istanze si aggiornano perché i nomi delle voci fanno parte del tipo di dato.

Un'istanza di una definizione di tipo può avere la propria didascalia, etichetta, descrizione, tip strip, valore di default, dimensione, colore o stile del controllo o indicatore, come ad esempio una manopola invece di un cursore slide.

Se cambiate il tipo di dato in una definizione di tipo, LabVIEW converte il vecchio valore di default nelle istanze della definizione di tipo con il nuovo tipo di dato, se possibile. LabVIEW non può preservare il valore di default dell'istanza se il tipo di dato cambia in un tipo incompatibile, come nella sostituzione di un controllo o di un indicatore numerico con un controllo o un indicatore stringa. Quando il tipo di dato di una definizione di tipo cambia in un tipo di dato incompatibile con la precedente definizione di tipo, LabVIEW imposta il valore di default delle istanze sul valore di default che avete specificato nel file `.ctl`. Se non specificate un valore di default, LabVIEW utilizza il valore di default per il tipo di dato. Per esempio, se cambiate una definizione di tipo da un tipo numerico ad un tipo stringa, LabVIEW sostituisce ogni valore di default associato con il vecchio tipo di dato numerico con stringhe vuote.

Definizioni di tipo Strict

Una definizione di tipo strict forza tutto ciò che è relativo ad una istanza ad essere identico alla definizione di tipo strict, eccetto la didascalia, l'etichetta, la descrizione, la tip strip e il valore di default. Come nel caso delle definizioni di tipo, il tipo di dato di una definizione di tipo strict rimane la stessa dovunque utilizzate la definizione di tipo strict. Le definizioni di tipo strict definiscono anche altri valori, come la verifica dell'intervallo nei controlli numerici e i nomi delle voci nei controlli ring. Le sole proprietà del VI Server disponibili per le definizioni di tipo strict sono quelle che riguardano l'aspetto del controllo o indicatore, come Visible, Disabled, Key Focus, Blinking, Position e Bounds.

Non potete impedire ad una istanza di una definizione di tipo strict di aggiornarsi automaticamente a meno che non rimuoviate il collegamento tra l'istanza e la definizione stessa.

Le definizioni di tipo e le definizioni di tipo strict creano un controllo personalizzato utilizzando un cluster di molti controlli. Se avete bisogno di aggiungere un nuovo controllo e di passare un nuovo valore ad ogni subVI, potete aggiungere il nuovo controllo al cluster dei controlli personalizzato. Questo evita di dover aggiungere il nuovo controllo al pannello frontale di ogni subVI e di realizzare nuovi collegamenti e terminali.

Per fare pratica con i concetti di questa sezione, completate l'Esercizio 5-3.

Esempio

Autovalutazione: Quiz

1. Potete creare un array di array.
 - a. Vero
 - b. Falso
2. Avete due array in ingresso collegati ad un For Loop. L'auto-index è abilitato su entrambi i tunnel. Un array ha 10 elementi, il secondo ne ha 5. Un valore di 7 è collegato al Count terminal, come mostrato in Figura 5-17. Qual è il valore dell'indicatore Iterations dopo l'esecuzione di questo VI?

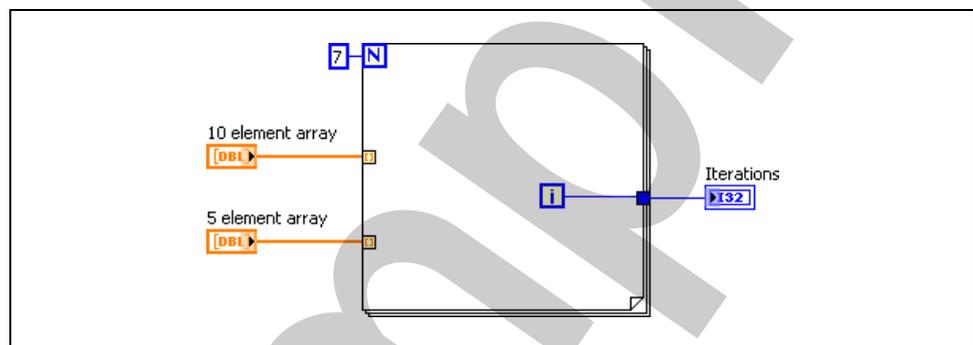


Figura 5-17. Qual è il valore dell'indicatore Iterations?

3. Personalizzate un controllo, selezionate **Control** dal menu a tendina **Control Type** e salvate il controllo come un file `.ctl`. Utilizzate quindi un'istanza del controllo personalizzato sul vostro pannello frontale. Se aprite il file `.ctl` e modificate il controllo, cambierà il controllo sul pannello frontale?
 - a. Sì
 - b. No
4. State inserendo dati che rappresentano un cerchio. I dati del cerchio comprendono tre numeri a doppia precisione: posizione x, posizione y e raggio. In futuro, potreste aver bisogno di espandere tutte le istanze dei dati del cerchio per includere il colore del cerchio, rappresentato come un intero. Come rappresentereste il cerchio sul vostro pannello frontale?
 - a. Tre controlli separati per le due posizioni e il raggio.
 - b. Un cluster contenente tutti i dati.
 - c. Un controllo personalizzato contenente un cluster.
 - d. Una definizione di tipo contenente un cluster.
 - e. Un array con tre elementi.

Esempio

Autovalutazione: Risposte ai quiz

1. Potete creare un array di array.
 - a. Vero
 - b. Falso**

Non potete trascinare un tipo di dato array in un contenitore array. Tuttavia, potete creare array bidimensionali.
2. Avete due array in ingresso collegati ad un For Loop. L'auto-index è abilitato su entrambi i tunnel. Un array ha 10 elementi, il secondo ne ha 5. Un valore di 7 è collegato al Count terminal, come mostrato nella figura seguente. Qual è il valore dell'indicatore Iterations dopo l'esecuzione di questo VI?

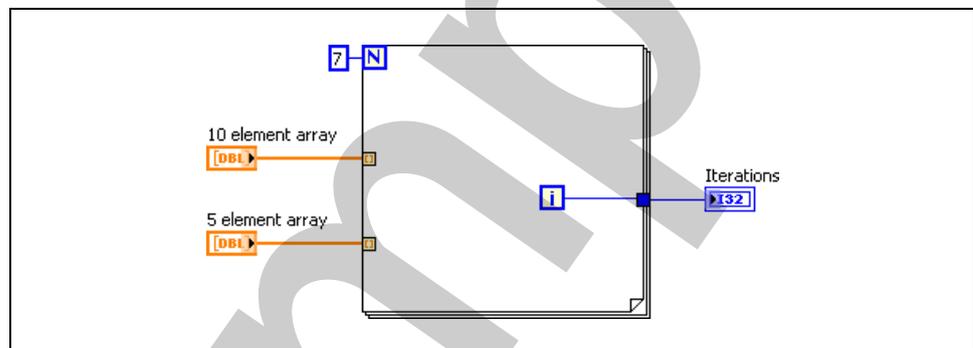


Figura 5-18. Qual è il valore dell'indicatore Iterations?

Valore di Iterations = 4

LabVIEW non supera le dimensioni dell'array. Questo aiuta a proteggere contro gli errori di programmazione. Le funzioni matematiche di LabVIEW lavorano nello stesso modo—se collegate un array a 10 elementi all'ingresso *x* della funzione Add e un array a 5 elementi all'ingresso *y* della funzione Add, l'uscita è un array a 5 elementi.

Sebbene il ciclo for venga eseguito per 5 volte, le iterazioni partono da zero, quindi il valore degli indicatori Iterations è 4.

3. Personalizzate un controllo, selezionate **Control** dal menu a tendina **Control Type** e salvate il controllo come un file `.ctl`. Utilizzate quindi un'istanza del controllo personalizzato sul vostro pannello frontale. Se aprite il file `.ctl` e modificate il controllo, cambierà il controllo sul pannello frontale?
 - a. Sì
 - b. No**

4. State inserendo dati che rappresentano un cerchio. I dati del cerchio comprendono tre numeri a doppia precisione: posizione x, posizione y e raggio. In futuro, potreste aver bisogno di espandere tutte le istanze dei dati del cerchio per includere il colore del cerchio, rappresentato come un intero. Come rappresentereste il cerchio sul vostro pannello frontale?
 - a. Tre controlli separati per le due posizioni e il raggio.
 - b. Un cluster contenente tutti i dati.
 - c. Un controllo personalizzato contenente un cluster.
 - d. Una definizione di tipo contenente un cluster.**
 - e. Un array con tre elementi.

ESempio

Nota

Esempio

Nota

Esempio