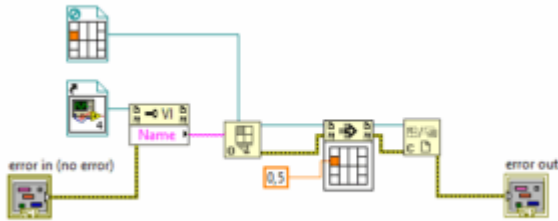


Caricamento Dinamico VI in LabVIEW

Category: LabVIEW Talk, News

scritto da Davide Oratelli | Febbraio 12, 2025



In LabVIEW, le chiamate dinamiche dei VI (Virtual Instruments) consentono di caricare ed eseguire VI in modo flessibile durante l'esecuzione dell'applicazione, offrendo vantaggi significativi in termini di modularità e gestione della memoria.

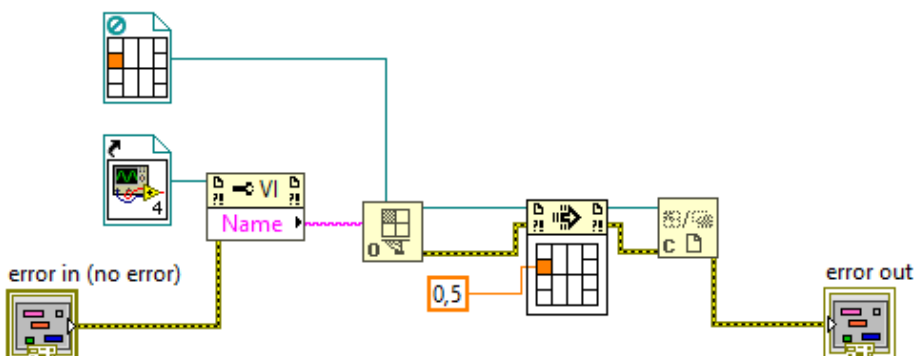
Questa tecnica è particolarmente utile in applicazioni complesse che richiedono l'esecuzione condizionale di moduli o l'estensione delle funzionalità tramite plugin.

Chiamata Sincrona tramite "Call By Reference"

Questa modalità utilizza il nodo 'Call By Reference' per eseguire un VI specificato da un riferimento strettamente tipizzato. Il VI chiamato viene eseguito in modo sincrono, bloccando il flusso di dati fino al completamento dell'esecuzione.

Per implementare questa tecnica:

- Utilizzare la funzione 'Open VI Reference' per ottenere un riferimento al VI target.
- Collegare il riferimento al nodo 'Call By Reference'.
- Fornire i dati di input necessari e gestire gli output.



Chiamata Asincrona tramite “Start Asynchronous Call”

Questa tecnica consente al VI chiamante di continuare l'esecuzione senza attendere il completamento del VI chiamato (Call and Forget).

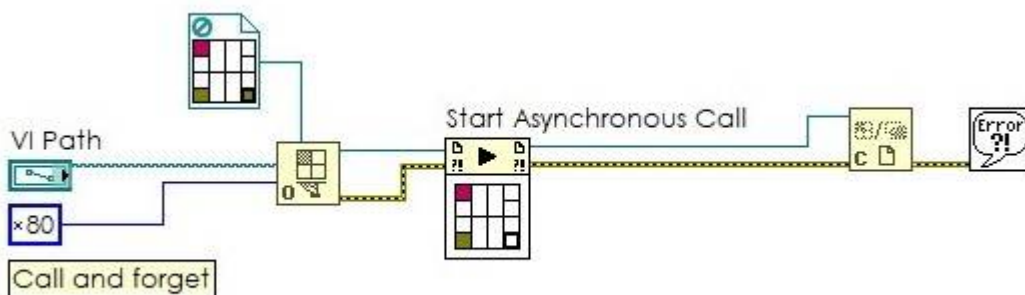
Esistono due principali modalità di chiamata :

Call and Forget

Il VI chiamato viene avviato e il chiamante non attende né raccoglie i risultati dell'esecuzione.

Per implementare questa modalità:

- Impostare il terminale 'options' della funzione 'Open VI Reference' con il valore 0x80.
- Utilizzare il nodo 'Start Asynchronous Call' per avviare il VI.

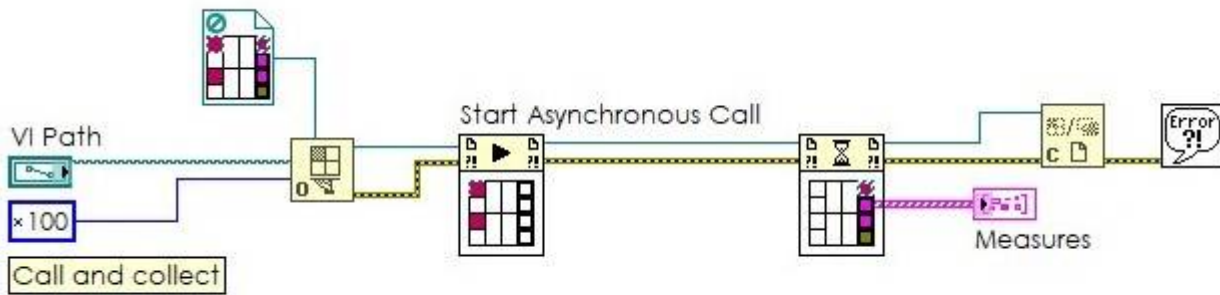


Call and Collect

Il VI chiamato viene avviato e il chiamante può successivamente raccogliere i risultati dell'esecuzione.

Per implementare questa modalità:

- Impostare il terminale 'options' della funzione 'Open VI Reference' con il valore 0x100.
- Utilizzare il nodo 'Start Asynchronous Call' per avviare il VI.
- Successivamente, utilizzare 'Wait on Asynchronous Call' per raccogliere i risultati.

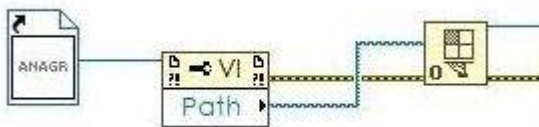


Riferimento Statico del VI

Un'alternativa al caricamento dinamico consiste nell'utilizzare un riferimento statico al VI. Questo approccio non richiede di specificare il percorso del VI, ma utilizza direttamente un riferimento statico.

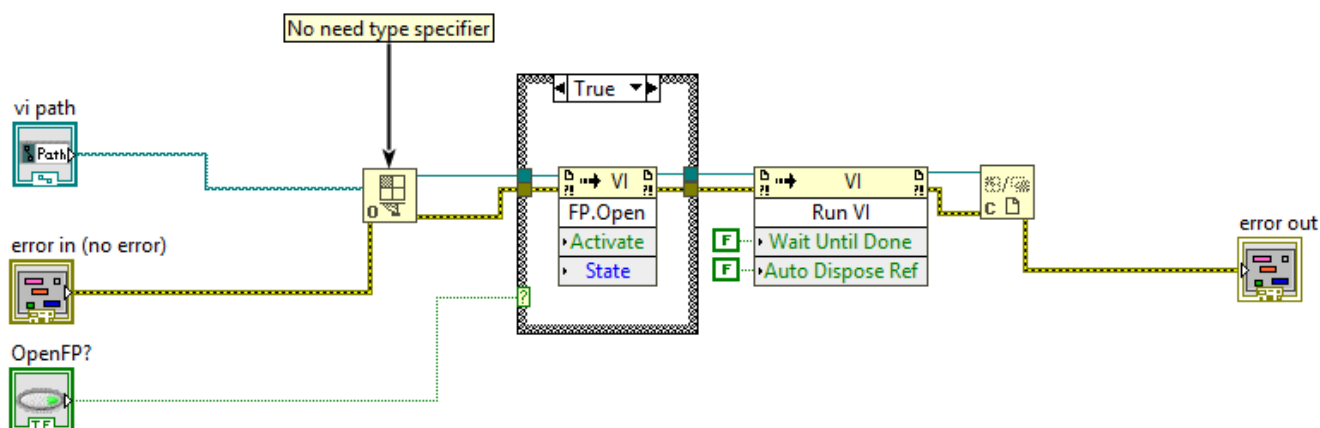
Passaggi per implementarlo:

- Posizionare una costante 'Static VI Reference' sul diagramma a blocchi.
- Utilizzare la proprietà 'VI Path' per ottenere il path del VI.
- Aprire il riferimento al VI utilizzando 'Open VI Reference' basandosi sul percorso ottenuto.
- Richiamare il metodo riferito al VI aperto chiamato "Run VI"



Chiamata con metodo Run VI

Il metodo Run VI consente di avviare l'esecuzione di un VI in modo simile al pulsante "RUN", utilizzando i valori correnti di tutti i controlli del pannello frontale anziché dati passati tramite parametri.



Questo approccio, che ignora le proprietà **Execution:Show Front Panel On Call** e **Execution:Close After Call**, offre una maggiore flessibilità: non è necessario utilizzare un riferimento strettamente tipizzato, il che permette di eseguire qualsiasi VI, anche quelli che non condividono lo stesso pannello connettore.

Inoltre, il metodo **Run VI** consente di interagire con il VI mentre è in esecuzione, ad esempio per monitorare o leggere ripetutamente gli output.

Tuttavia, se si desidera passare valori al VI, è necessario conoscere i nomi dei controlli e i relativi tipi di dati, il che comporta una programmazione aggiuntiva.

Pro

- Non richiede un riferimento strettamente tipizzato, permettendo di passare qualsiasi riferimento VI.
- Avvia il VI utilizzando i valori correnti dei controlli del pannello frontale.
- Consente l'interazione in tempo reale durante l'esecuzione, simile al pulsante "Esegui".

Contro

- Per passare valori è necessario conoscere i nomi e i tipi di dati dei controlli, il che richiede una programmazione aggiuntiva.
- Se si utilizza il nodo Call by Reference, i valori vengono emessi solo al termine dell'esecuzione, limitando l'interazione.

Tabella di confronto

Caratteristiche	Run VI	Start Asynchronous VI	Wait on Asynchronous Call	Chiamata tramite SubVI
Tipo di esecuzione	Avvia il VI immediatamente, similmente al pulsante "Esegui".	Avvia il VI in modalità asincrona, permettendo l'esecuzione in background senza bloccare il flusso del chiamante.	Attende il completamento del VI avviato in modalità asincrona, bloccando il flusso del chiamante fino alla fine dell'esecuzione.	Esecuzione sincrona: il VI viene chiamato direttamente come parte del flusso del codice, con esecuzione inline.

Gestione dei parametri e dati	Utilizza i valori correnti del front panel; per passare valori è necessario conoscere nomi e tipi dei controlli.	Può essere configurato per utilizzare parametri o i valori correnti del front panel, in base alla logica implementata nel VI.	Recupera gli output dal VI una volta completata l'esecuzione asincrona, per poi elaborare i dati risultanti.	I parametri vengono passati tramite wiring (connettore) in maniera strettamente tipizzata, garantendo il controllo a compile-time.
Interazione con il VI	Consente l'interazione diretta con il front panel durante l'esecuzione, permettendo aggiornamenti in tempo reale.	L'esecuzione avviene in background, non offrendo un'interazione diretta durante l'esecuzione del VI.	Non consente interazioni fino al termine del VI asincrono, in quanto il flusso resta in attesa per sincronizzarsi.	Il front panel del VI non viene mostrato durante l'esecuzione; l'interazione avviene esclusivamente tramite i dati passati in ingresso e restituiti in uscita.
Uso tipico	Ideale per avviare dinamicamente un VI quando si desidera interagire in tempo reale con il front panel e utilizzare i valori correnti.	Utile in applicazioni che richiedono esecuzioni concorrenti o non bloccanti, dove il VI deve operare in background.	Utilizzato per sincronizzare il flusso di esecuzione, aspettando il completamento di un VI avviato asincronicamente per procedere con i dati.	Indicato per chiamate statiche di VI, quando si necessita di un controllo rigoroso dell'interfaccia e dei tipi di dati grazie alla connessione strettamente tipizzata.
Note aggiuntive	Ignora le proprietà <i>Execution:Show Front Panel On Call</i> e <i>Execution:Close After Call</i> .	Consente di gestire VI in parallelo grazie al token asincrono, ma richiede una gestione specifica della sincronizzazione.	È strettamente legato al token restituito da Start Asynchronous VI e garantisce l'acquisizione corretta degli output al termine.	Fornisce il vantaggio della verifica a compile-time e una gestione integrata dell'interfaccia, sebbene non offra la flessibilità dinamica dei metodi Run VI.

Vantaggi del Caricamento Dinamico dei VI

- Modularità: facilita l'integrazione di componenti indipendenti.
- Gestione della memoria: carica i VI solo quando necessario.
- Flessibilità: consente di implementare architetture plugin.

Considerazioni e Limitazioni

- Gestione dei riferimenti: evitare memory leak gestendo correttamente i riferimenti.
- Sincronizzazione: necessario per evitare conflitti nell'accesso a risorse condivise.
- Debugging: più complesso rispetto ai VI statici.

Esempi Applicativi

- Caricamento Condizionale: esecuzione di VI solo quando necessario.
- Architetture Plugin: per espandere le funzionalità di un'applicazione.
- Gestione di Processi Paralleli: miglioramento delle prestazioni.