

Servizi web in LabVIEW

Soluzioni possibili, come si utilizzano.

1

Soluzioni possibili ... WEB SERVER

- Dalla versione 5.1 di LabVIEW è possibile implementare un **Web server** che consente di operare da remoto sul pannello frontale di una macchina usando un comune browser.
- Il pannello frontale del VI (Virtual Instrument) viene caricato come una **pagina html dinamica** della quale deve essere fornito l'URL (Uniform Resource Locator: sequenza di caratteri per mezzo della quale viene identificata univocamente la risorsa in Internet).
 - Esempio: per operare sul pannello frontale del VI: 'test.vi' in esecuzione su una macchina avente indirizzo IP 127.0.0.1, è necessario inserire nel browser l'URL: <http://127.0.0.1/test.html>. (L'indirizzo IP 127.0.0.1 è un indirizzo molto particolare che rappresenta il localhost e viene comunemente usato per testare applicativi web.)

2

WEB SERVER

- Per controllare un'applicazione LabVIEW agendo da remoto sul pannello frontale mediante un Browser è necessario:
 - Impostare il Web Server (in particolare è necessario abilitare sia l'accesso in lettura che in controllo);
 - Eseguire il VI da controllare;
 - Accedere alla pagina html corrispondente al pannello frontale da comandare.
 - Se nessun altro client ha già il controllo del VI, il controllo viene fornito al client in esame, altrimenti la richiesta viene accodata alle altre richieste ricevute ed il servizio viene fornito quando tutte le precedenti richieste sono state soddisfatte.

3

Web Publishing Tool

- Consente di pubblicare il pannello frontale di un VI, ovvero di creare una **pagina html contenente le immagini** sia statiche che animate **degli oggetti del pannello frontale**.
- Le immagini degli oggetti del pannello frontale vengono salvate nei formati jpeg oppure png.
 - Usando l'opzione embedded
 - Usando l'opzione **snapshot** il pannello frontale viene convertito in un insieme di immagini statiche;
 - Usando l'opzione **monitor** invece il pannello frontale viene convertito in un insieme di immagini animate.
- **IMPORTANTE:** la dimensione del pannello frontale influisce significativamente sulla dimensione delle immagini da creare e quindi sul tempo necessario al Web Server per creare l'immagine e sul tempo necessario al Browser per caricare l'immagine (tempo di trasmissione).

4

CLIENT

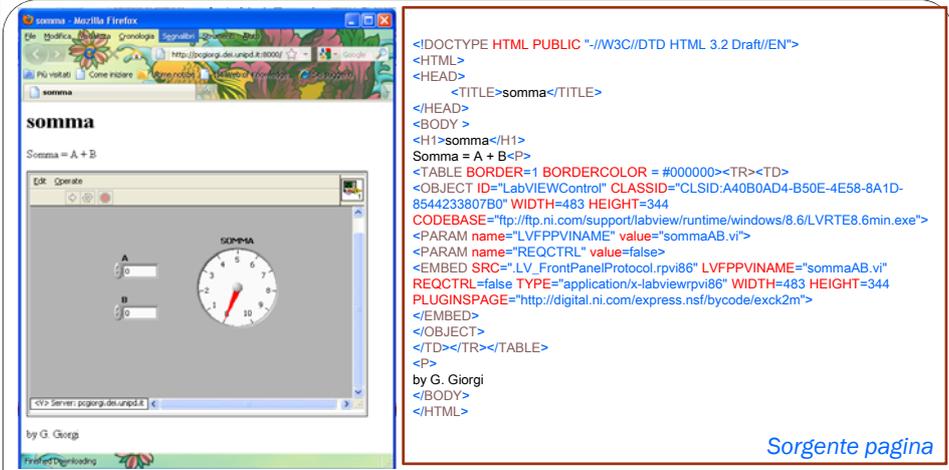
- Per accedere al pannello remoto pubblicato come descritto è possibile utilizzare un client in Labview, oppure un comune Web Browser (opzione che non richiede l'installazione di LabVIEW).
- Nel caso di un comune Web Browser, è comunque **necessario installare LabVIEW Run-Time Engine** (fornito gratuitamente dalla National Instrument).

5

Avvertenze

- Pannelli frontali contenenti controlli ed indicatori complessi (dialog boxes, ecc.) possono non funzionare correttamente;
- Evitare di esportare VI aventi cicli while privi di attesa;
- Controlli .NET e ActiveX non funzionano correttamente;
- Si possono avere problemi con i Property Node, ecc.

6



by G. Giorgi

[Sorgente pagina](#)

Esempio

Publicazione di un VI che esegue la somma tra due numeri A e B

URL: <http://pcgiorgi.dei.unipd.it:8000/sommaAB.html>

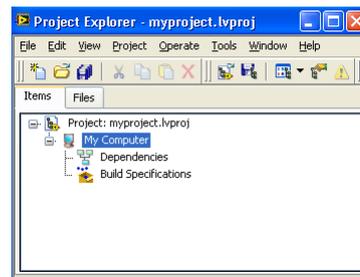
7

Soluzioni possibili ... Web Services

- Con LabVIEW 8.6 è stata resa disponibile una diversa modalità per la gestione remota di VI. Questa modalità prende il nome di **Web Services** e si basa su concetti completamente differenti da quelli utilizzati per la creazione di pannelli remoti.
- I vantaggi offerti da questa diversa soluzione sono molteplici; i più importanti sono:
 - È possibile **comunicare** con applicazioni Labview **da qualsiasi dispositivo in grado di connettersi al Web** (anche un telefono cellulare o un microcontrollore dotato di interfaccia di rete -Ethernet, ZigBee, BlueTooth, WiFi, ecc.-);
 - La comunicazione avviene usando il **protocollo HTTP** e non un protocollo proprietario (non è più necessaria l'installazione di LabVIEW Run-Time Engine);
 - Applicazioni LabVIEW possono essere controllate da remoto usando **client personalizzati** (custom client), come ad esempio quelli creati con l'open source framework *Adobe Flex* <http://www.adobe.com/products/flex/overview/>;
 - E' possibile gestire **qualsiasi tipo di dato MIME** (Multipurpose Internet Mail Extensions) , così come testi, immagini e video.

Configurazione di un Web Services

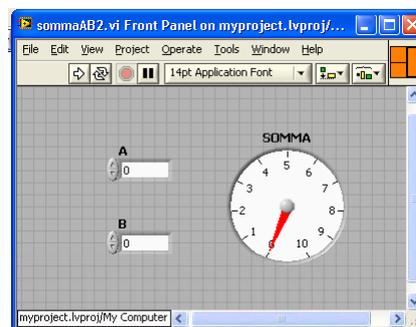
- Per configurare un VI in modo tale da poter essere gestito da remoto, è necessario come prima cosa inserire il VI in un progetto:
 - Dalla pagina iniziale di LabVIEW (**Getting started**) creare un progetto vuoto (opzione **Empty Project**) e salvarlo con il nome desiderato (esempio: myproject.lvproj).
 - Il progetto creato viene gestito attraverso i meccanismi disponibili nella sezione **Project Explorer** (che viene aperta automaticamente alla creazione di un nuovo progetto).
 - Cliccando con il tasto destro del mouse su **MyComputer** si apre un menù a tendina. Scegliere **New→VI** per aggiungere un nuovo VI al progetto.



9

Configurazione di un Web Services

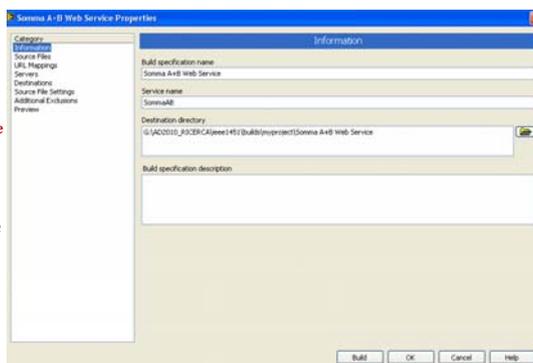
- Si consideri per semplicità l'esempio precedente dove venivano sommati due numeri, A e B:
 - Creare, per il nuovo VI, un pannello frontale con due controlli A e B di tipo double ed un indicatore, chiamato SOMMA sempre di tipo double.
 - Collegare i pin del connettore (basta un connettore a 3 pin) rispettivamente ai due controlli e all'indicatore.
 - Nel block diagram inserire il codice necessario per eseguire la somma tra A e B ed inviare il risultato all'indicatore SOMMA.
 - Salvare il VI (esempio: sommaAB2.vi)



10

Configurazione di un Web Services

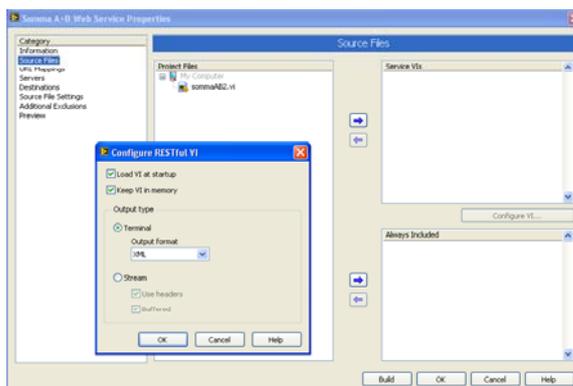
- Creare quindi un Web Services: cliccare con il tasto destro del mouse su **Build Specification** e scegliere **New→Web Service (RESTful)**.
- Con la creazione del Web Service viene aperta la finestra **My Web Service Properties** mediante la quale configurare il servizio.
 - La prima informazione da specificare è il **Build Specification Name** che associa un nome al Web Services che si sta costruendo (il nome viene usato per creare la directory di destinazione, **Destination Directory**).
 - La seconda informazione è il **Service Name** che viene usato per costruire l'URL mediante il quale identificare la risorsa in internet (vedere la costruzione dell'URL).



11

Configurazione di un Web Services

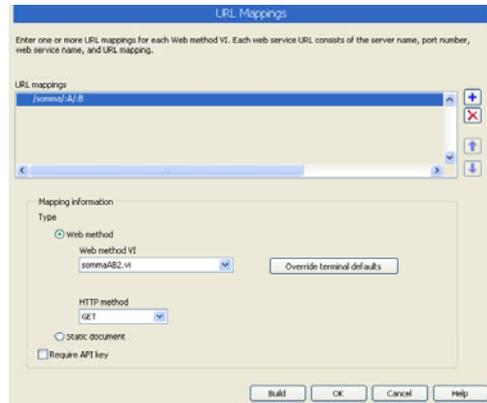
- Dopo aver inserito le informazioni nel sottomenù **Information**, accedere al sottomenù **Source Files**.
- Dal menù **Source Files** si possono selezionare i VI da associare al Web Services.
- E' possibile configurare le proprietà del servizio per ciascun VI selezionato. Da notare la possibilità di configurare il tipo di uscita prodotta dal VI:
 - da terminale (come nell'esempio dove il terminale di uscita corrisponde a SOMMA). E' necessario scegliere il formato dell'uscita (text, html, xml)
 - Da stream (si vedrà più avanti come usare questa opzione).



12

Configurazione di un Web Services

- Accedere infine al sottomenù **URL mappings** mediante il quale associare ad ogni VI, eseguito come Web Service, il corrispondente URL.
- Nell'esempio **somma** è il nome scelto per il servizio.
- Per collegare gli ingressi dell'URL ai terminali di ingresso del VI è necessario inserire la stringa "**nome_terminale**" alla fine dell'URL. Ogni terminale deve essere separato con una barra **"/**".
- Per inviare i dati al VI è necessario infine specificare il modo in cui tali dati vengono passati. I metodi più comuni definiti dallo standard HTTP sono GET e POST.
- In questo esempio viene usato il metodo GET.



13

Configurazione di un Web Services

- Dopo queste impostazioni, chiudere la finestra **My Web Service Properties** cliccando su OK.
- Dalla finestra **Project Explorer**, cliccare con il tasto destro del mouse sul Web Service creato (nel caso dell'esempio **Somma A+B Web Service**), e compilare mediante l'opzione **Build**.
- Dopo aver compilato è necessario distribuire il servizio. Cliccare sul Web Service creato con il tasto destro del mouse e scegliere l'opzione **Deploy**.
- Avviare un browser. Per eseguire la somma tra due numeri, ad esempio 3.12 e 6.43 usare il seguente comando:

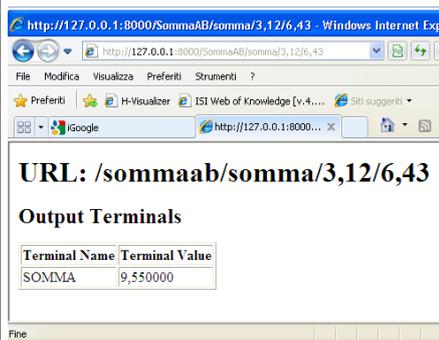
<http://127.0.0.1:8000/SommaAB/somma/3,12/6,43>

14

Esempio

- <http://127.0.0.1:8000/SommaAB/somma/3,12/6,43>

Sorgente pagina



```
<html>
<body>
<h1>URL: /sommaab/somma/3,12/6,43</h1>
<h2>Output Terminals</h2>
<table border=1.0>
<tr>
<th>Terminal Name</th>
<th>Terminal Value</th>
</tr>
<tr>
<td>SOMMA</td>
<td>9,550000</td>
</tr>
</table>
</body>
</html>
```

15

Richiami HTTP

- Le funzionalità appena presentate sono per molti aspetti limitate e un migliore e più efficiente utilizzo delle potenzialità offerte da LabVIEW Web Services si può raggiungere.
- A tal fine però è bene fare un **breve richiamo sul protocollo HTTP** (Hypertext Transfer Protocol).

16

Richiami HTTP

- HTTP è uno tra i protocolli più utilizzati per trasferire informazioni sul Web.
- Si basa su un protocollo di tipo client-server che consente a due processi di comunicare usando una connessione TCP/IP.
- Un **server HTTP** è un programma, che risiede in un dato host, e che rimane in ascolto di eventuali richieste su una porta prestabilita (tipicamente 80 per sistemi Windows; nell'esempio proposto è stata scelta la porta 8000 in quanto la porta 80 era già usata da un altro HTTP server).
- Un **client HTTP** è un programma che richiede l'apertura di una connessione TCP/IP al server attraverso un socket, trasmette una richiesta e rimane in attesa di una risposta dal server (un comune Browser è un client HTTP).
- Il protocollo HTTP definisce 8 metodi mediante i quali il client può specificare l'azione desiderata da eseguire sulla risorsa Web. Questi metodi sono: GET, POST, HEAD, PUT, DELETE, TRACE e OPTIONS.
- I metodi più usati tuttavia sono solamente due: **GET** e **POST**.

17

Richiami HTTP: GET e POST

- Il **metodo GET** consente di recuperare informazioni dal server. L'interazione tra client e server è simile ad una domanda, come può essere una richiesta (query), un'operazione di lettura o di controllo.
- Tipicamente GET viene usato per recuperare dati dal server.
- Il **metodo POST** viene usato per richiedere al server di accettare l'entità inclusa nella richiesta come nuova condizione per la risorsa individuata dall'URL.
- Tipicamente POST viene usato per invio di dati complessi (esempio la sottoscrizione di un servizio).

18

Richiami HTTP: sintassi

- La sintassi usata da HTTP per identificare una risorsa nel Web (ovvero la sintassi della stringa URL) è specificata dalla raccomandazione RFC2616:

`http://<host>:<port>/<path>?<parameters>`

- <host>** identifica il dispositivo su cui è in esecuzione il server (es. indirizzo IP del computer);
- <port>** è il numero della porta associata a tale servizio (per default il numero è 80 ma può essere cambiato se tale valore è già assegnato ad un'altra risorsa);
- <path>** è il percorso che identifica il server all'interno dell'host;
- <parameters>** è la lista dei parametri associati alla risorsa. Per ogni parametro deve essere specificato l'identificativo (key) ed il valore separati dal carattere "=". Per separare i vari parametri viene usato il carattere "&". Esempio: "A=3,12&B=6,43"

19

Esempio

- Si vuole realizzare un VI con le stesse funzionalità del precedente.
- La riga di comando usata per interagire con questo VI è però una query del tipo:

`http://<host>:<port>/<path>?A=3,12&B=6,43`

Il VI ha un solo terminale associato all'unico controllo **HttpRequestID**. Questo controllo deve essere connesso ad uno dei pin del connettore.

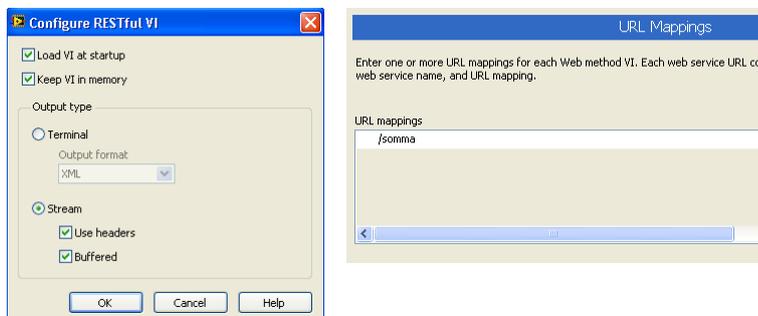
I parametri **A** e **B** specificati nella riga di comando (dopo il **?**) vengono letti per mezzo del VI **Read Form Data**, che restituisce una stringa per ciascun valore.

I valori associati ai parametri **A** e **B** vengono convertiti in numeri, sommati, riconvertiti in stringa ed inviati al buffer in uscita gestito dal VI **Write Response**.

20

Il formato della risposta inviata al client viene impostato mediante il VI **Set HTTP Response MIME Type**. Nell'esempio viene scelto il formato **html**.

Esempio



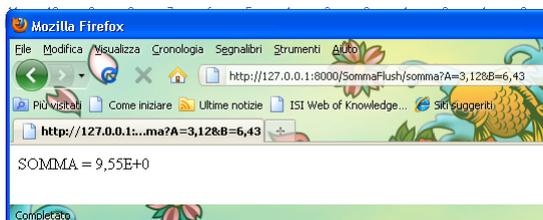
Rispetto al caso precedente l'uscita del VI deve essere di tipo **Stream**.
Non essendoci terminali di ingresso in grado di ricevere valori da remoto, l'URL è composto dal solo nome della risorsa.

N.B.: L'unico terminale di ingresso è **httpRequestID** (connesso ad uno dei PIN del connettore) che viene inizializzato con l'identificativo della sessione HTTP in corso.

21

Esempio

- <http://127.0.0.1:8000/SommaFlush/somma?A=3,12&B=6,43>



Sorgente pagina

SOMMA = 9,55E+0

22