

Data 03/04/2025

Connettore MagTrace – Summa.

**Guida completa per lo sviluppo / utilizzo del software.**

Il progetto del Connettore MagTrace è chiamato EsportaDBF e si trova sul Nas sotto la cartella backup Gabriele > Sorgenti.

Il connettore deve leggere dai file DBF (File database che interagiscono con il gestionale Summa) i dati degli articoli, clienti, fornitori, e scrivere le fatture all’interno di essi.

Per eseguire queste operazioni vengono utilizzati dei programmi in C++ che convertono il file da DBF a CSV nel caso della lettura o scrivono in modalità binaria su di essi.

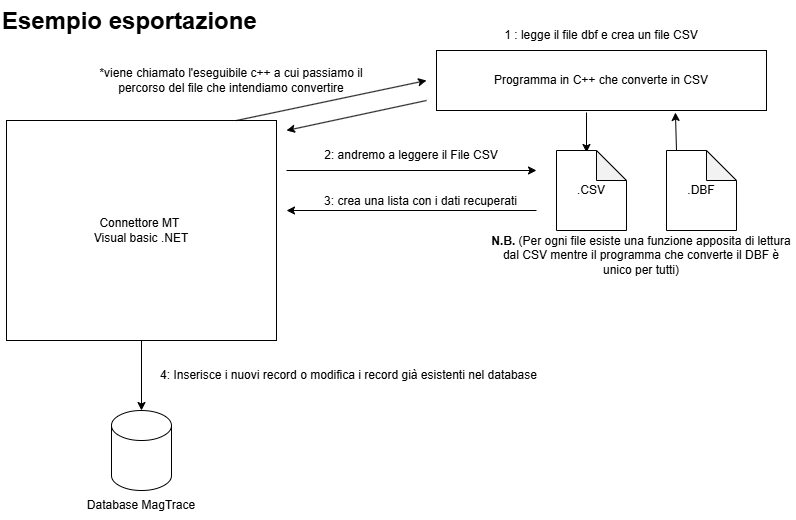
Ad ogni operazione noi copiamo il file , se per qualche motivo la copia non riesce l’operazione viene interrotta.

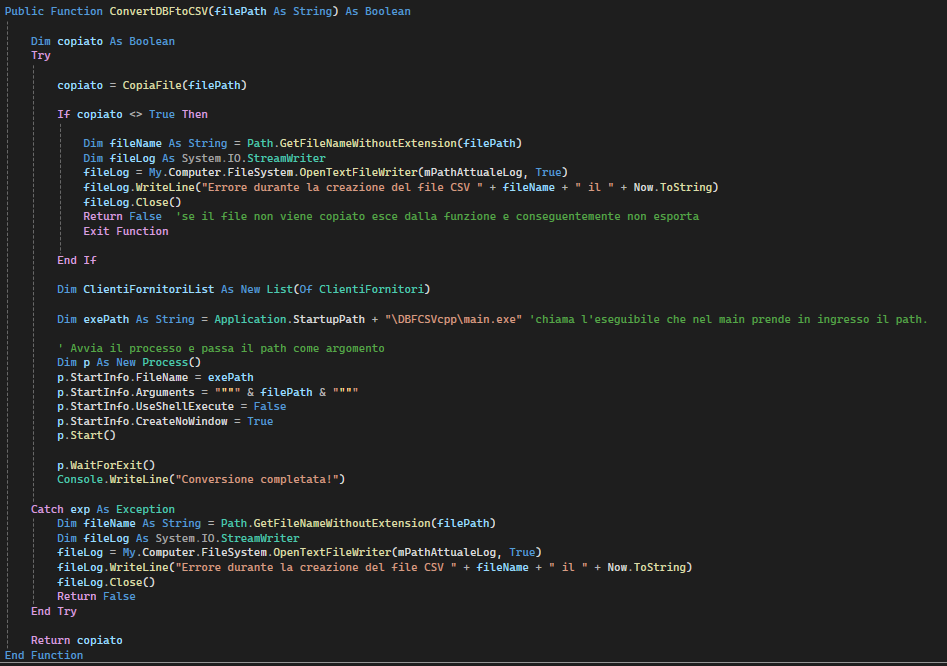
I programmi in C++ si trovano nella directory dell’eseguibile , ognuno di essi ha una cartella apposita e sono:

|  |  |
| --- | --- |
| Nome della cartella | Utilizzo |
| DBFCSVcpp | Serve per convertire i file DBF in CSV che verranno letti tramite apposite funzioni nel software in visual basic .net. |
| DBFIMPFattGencpp | Viene utilizzato per scrivere all’interno del file Fattgen (Ovvero il file di testata delle fatture), vengono passati gli argomenti di un oggetto fattura che sarà poi inserito. |
| DBFIMPFattRigcpp | Viene utilizzato per scrivere all’interno del file Fattrig (Ovvero il file di dettagli delle fatture che contiene gli articoli), vengono passati gli argomenti di un oggetto fattura che sarà poi inserito. |

**N.B. Per tutti questi il file da eseguire per l’importazione / esportazione è sempre main.exe.**

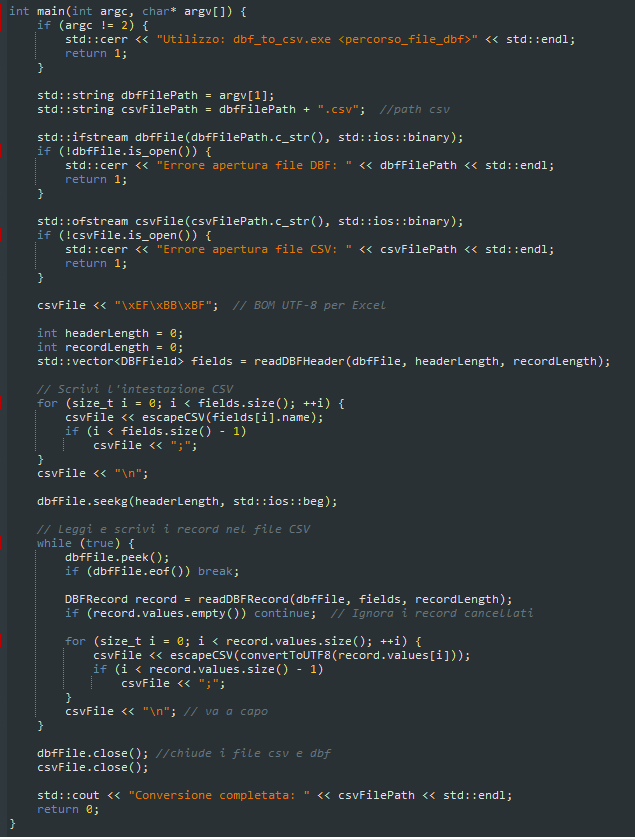
## Esportazione da DBF

**Funzione per convertire in CSV:** 

Come solo argomento diamo il percorso.

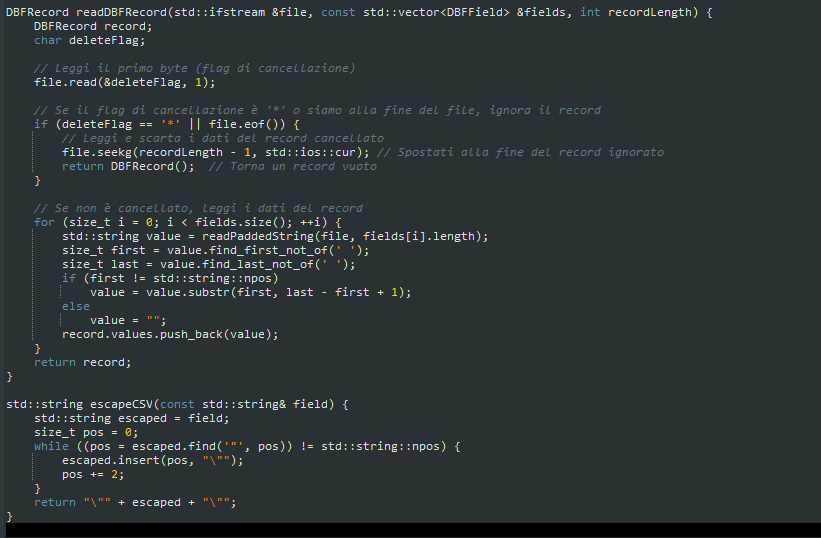
**N.B. Prima di convertire in CSV e prendere i dati viene fatta una verifica sulla data di modifica del file interessato , se non è stato modificato non viene fatta alcuna operazione nel caso dell’automatismo, possiamo comunque forzare l’aggiornamento tramite gli appositi pulsanti.**

**Funzione Main di DBFCSVcpp :**



Crea il file CSV e lo apre , allo stesso tempo apre il Dbf e cicla i dati , come prima cosa le intestazioni poi i record ignorando quelli cancellati. (con il flag \*)

**Funzioni readDBFRecord e escapeCSV di DBFCSVcpp:**



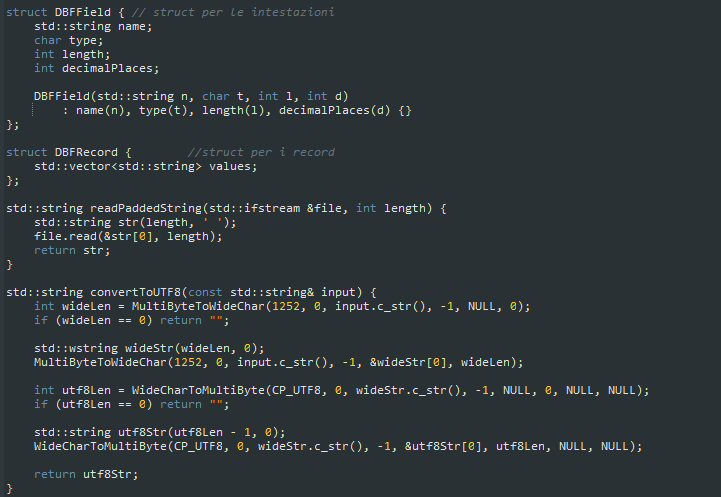
-La funzione readDBFRecord verifica se viene trovato il delete flag e ignora la riga se presente, se non cancellato legge i dati del record e lo ritorna.

-EscapeCSV serve per scrivere l'intestazione nel file CSV



readDBFHeader legge l’ intestazione e restituisce un vettore di DBFField.

**Struct di DBFCSVcpp:**

-DBFField e DBFRecord sono delle struct , per rendere facile il concetto di struct è come se creassimo una classe , il C++ è effettivamente un linguaggio orientato ad oggetti e permette la creazione di essi.

-DBFField(str::string) è il costruttore della classe.

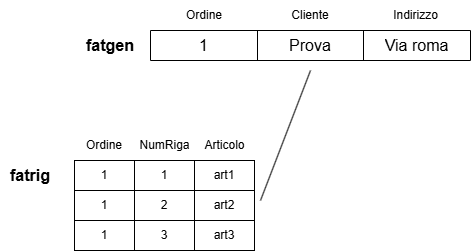
-readPadded aggiusta la stringa che viene presa aggiungendo degli spazi.

-convertToUTF8 converte le stringhe dal formato preso dal file dbf in formato UTF8.

## Importazione verso il file DBF

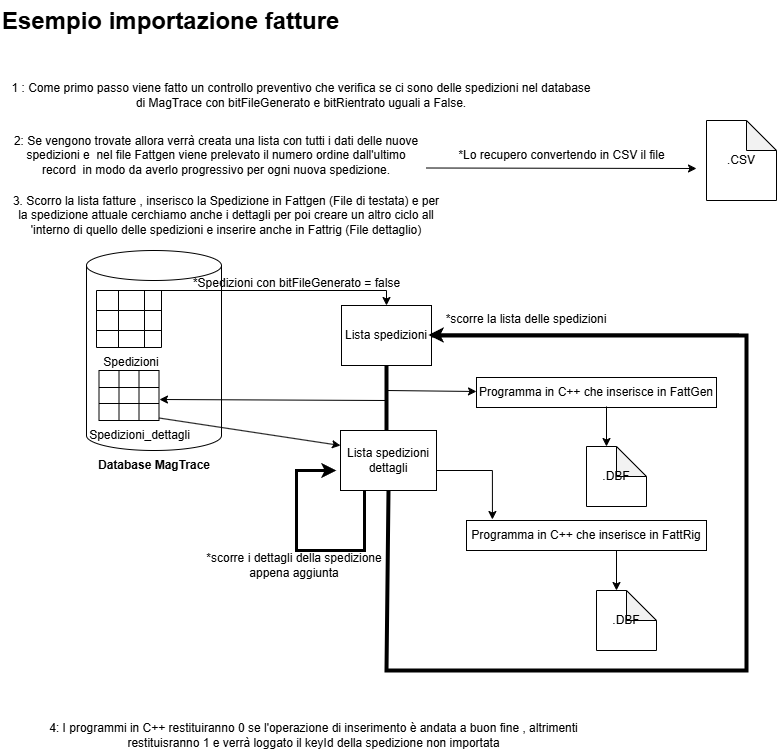
Per l’importazione verso summa abbiamo il tracciato che coincide tra gli argomenti di spedizioni e fattgen (Le testate) e spedizioni\_dettagli e fattrig (dettagli).

|  |  |
| --- | --- |
| **Spedizioni** | **Fattgen** |
|  | Ordine (Ultimo record + 1) |
| Nuo |  |
| Anno | (In fattrig) |
| Numero ordine cliente | nr\_ordi |
| Anno ordine cliente |  |
| Codice cliente | codice |
| codice destinazione cliente | Ragione Sociale Dest. merce, Indirizzo dest. merce,  Cap dest. merce, Comune dest. merce, Provincia dest. merce) da duplicare o lasciare vuoti nel caso di stessa destinazione Cliente |
| Data creazione | dataor |
| Ora creazione |  |
| Data consegna se prevista | Datac(?) |
| Ora consegna se prevista |  |
| Numero D.D.T | numdoc (??) |
| Numero colli | nr\_coll |
| Sezionale |  |

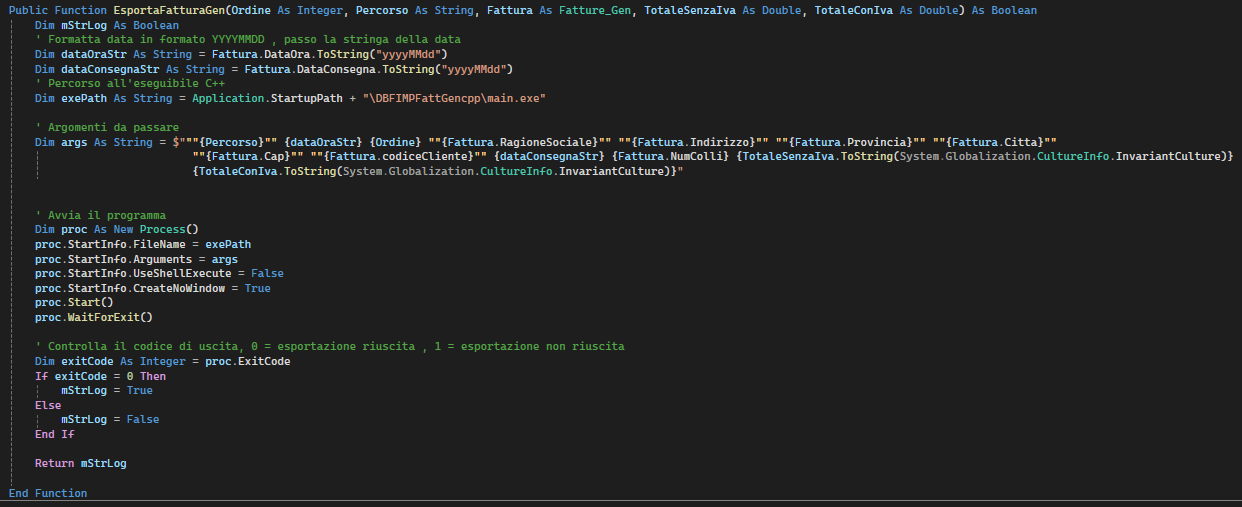


|  |  |
| --- | --- |
| **Spedizioni\_dettagli** | **Fattrig** |
|  | Anno (Da spedizioni) |
|  | Ordine (Da fattgen) |
|  | Cliente/Fornitore (Non utilizzato nel file) |
|  | Numero Riga (Progressivo dell’ articolo per ogni spedizione) |
| Codice articolo | cod |
| Descrizione articolo | des\_arti |
| Lotto interno prodotto | lotto |
| Quantità | N\_pezzi / quantita |
| Prezzo Unit | Pr\_unit |
| Unità misura vendita | Unita\_misu |
| Percentuale sconto 1 | Sconto |
| Percentuale sconto 2 | Sconto2 |
| Percentuale sconto 3 | Sconto3 |
| Flag omaggio esente IVA |  |
| Flag omaggio |  |
| Tracciabilita |  |

Meccanismo per l’ importazione:



Funzione per l’esportazione delle fatture generali:

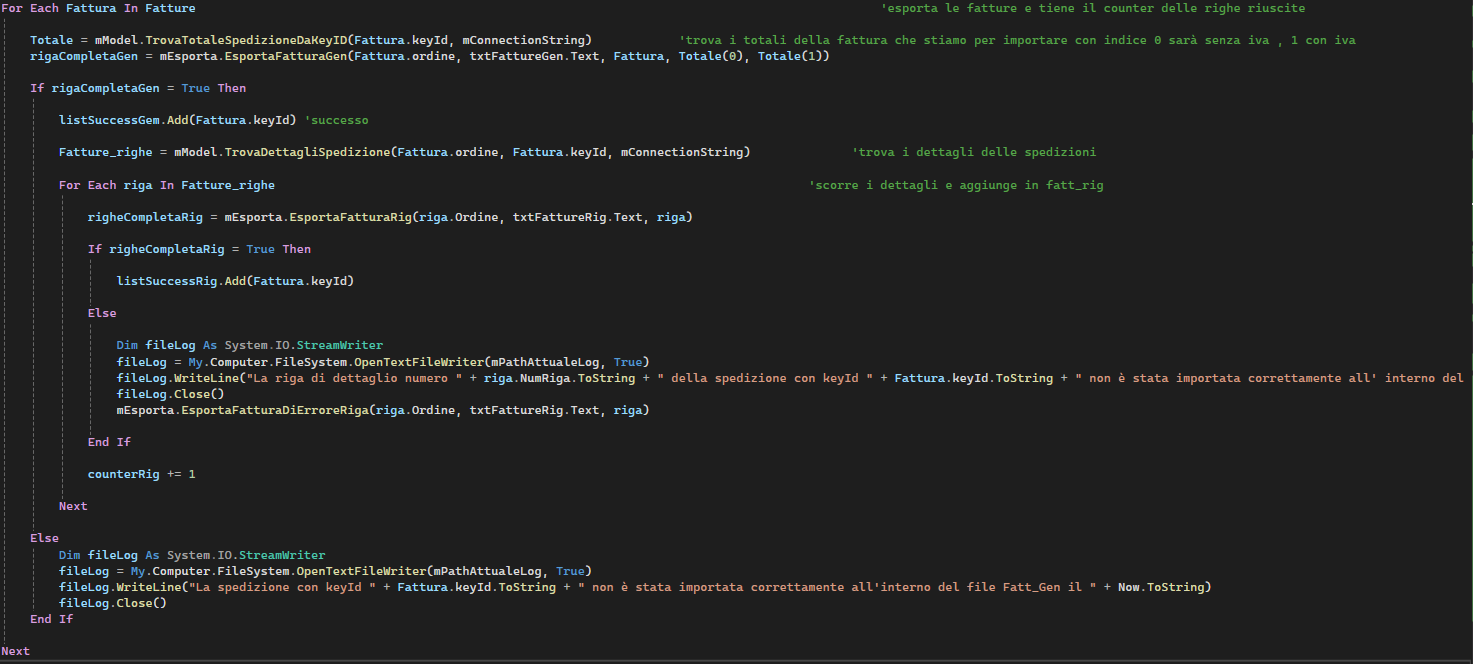


L’applicativo in C++ restituirà 0 se l’inserimento è riuscito o 1 se non è riuscito, nel caso delle righe (fattrig) di dettaglio verrà inserita una riga di “errore” per mostrare all’operatore che è avvenuto un errore.

Passiamo come argomenti i singoli parametri dell’oggetto che verrà poi inserito nel record.

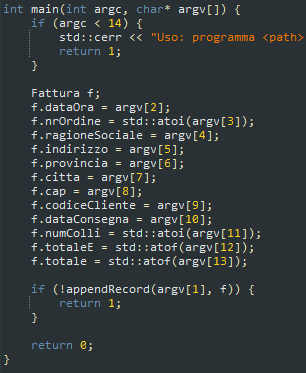
Stessi passaggi per fattrig ma con argomenti diversi ovviamente.

**Ciclo per l’inserimento delle fatture:**



Vengono recuperati i totali per poi essere inserite nel file DBF insieme all’oggetto fattura.

**Funzione Main di DBFIMPFattGencpp :**



Prendiamo gli argomenti in ingresso e li inseriamo all' interno di un oggetto fattura che poi manderemo nella funzione appendRecord() che prende in ingresso il path del file (quindi il primo argomento passato [1]) e l’oggetto fattura.

Nella funzione appendRecord() possiamo trovare il ciclo aggiunge i parametri dell’oggetto sotto le colonne del file DBF.



appendRecord restituirà false (di conseguenza 1 come exit code) se l’inserimento fallisce e true se riesce (0 come exit code).

Anche qui abbiamo le funzioni readDBFHeader e padField,

Inoltre abbiamo delle funzioni (intToString e doubleToString) che convertono i valori numerici in stringhe.

Il programma **DBFIMPFattRigCpp** è uguale , con le stesse funzioni , cambia solo la classe fatture che viene scambiata con una classe FatturaRiga.

## File XML per i settaggi

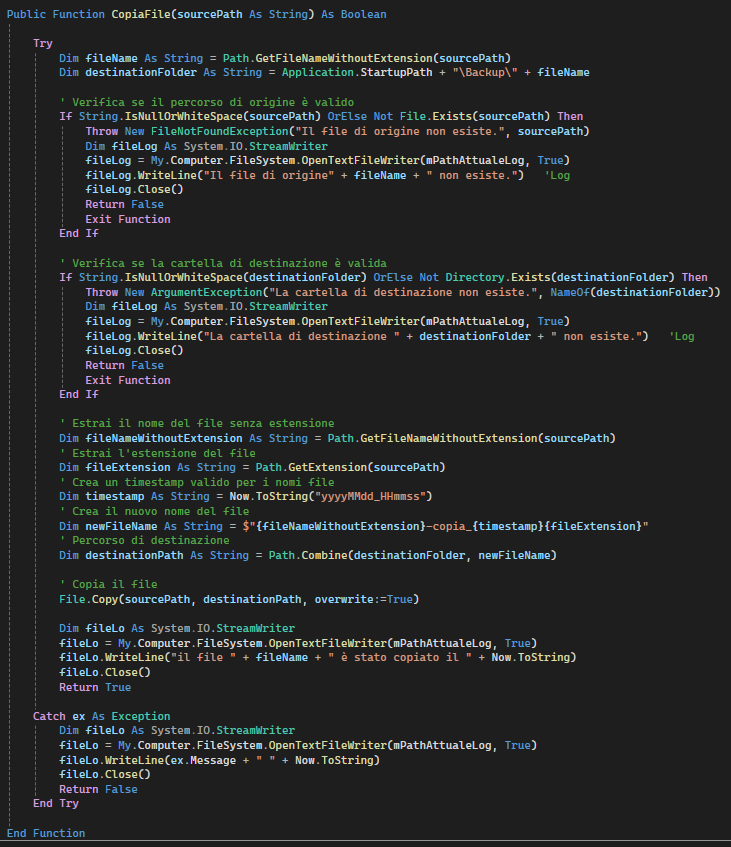
Il file che si trova sotto la directory dell’ eseguibile ‘ImportaDBF.exe.config’ sarà la fonte dei dati che popoleranno le textbox con i percorsi dei file dbf , il tempo di aggiornamento e di eliminazione dei file (di log e di backup)

Al suo interno sono presenti:

|  |  |
| --- | --- |
| **Key** | **Funzione** |
| Time | Variabile utilizzata per il tempo di aggiornamento delle esportazioni / importazioni (Espressa in secondi nel file , verrà poi convertita in minuti a runtime). |
| GiorniPerEliminazioneFile | Variabile utilizzata per esser confrontata con le date dei file di log e di backup, se un file è più vecchio della differenza tra data attuale e il numero di giorni inseriti esso viene eliminato. |
| DataArticoli | Data dell’ultima modifica del file DBF che contiene gli articoli |
| DataCliForn | Data dell’ultima modifica del file DBF che contiene clienti e fornitori |
| DataDestinazioni | Data dell’ultima modifica del file DBF che contiene le destinazioni dei clienti. |
| PathArticoli | Percorso del file DBF che contiene gli articoli. |
| PathDestinazioni | Percorso del file DBF che contiene le destinazioni clienti. |
| PathClientiFornitori | Percorso del file DBF che contiene clienti e fornitori. |
| PathFattGen | Percorso del file DBF per la testata delle fatture. |
| PathFattRig | Percorso del file DBF per i dettagli delle fatture |
| ConnectionString | Stringa di connessione al Database MagTrace |

## Copia dei file

Prima dell’esecuzione di ogni operazione i file coinvolti vengono copiati , la funzione per la copia è unica e copia il file all’interno della cartella Backup > nome del file.



Viene loggato sia se la copia va in l’errore sia se la copia viene creata con successo.

Se l’operazione di copia non va a buon fine l’operazione interessata non viene effettuata.

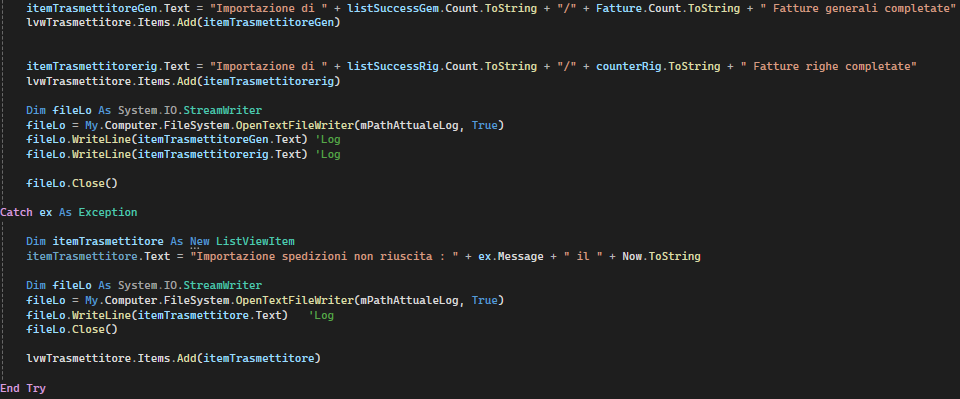
**Valori di ritorno : true se la copia è riuscita , false se fallisce.**

## Log e lista eventi

Ad ogni evento vengono loggati all’interno di un file di log con la data odierna lo stato delle operazioni.

Esempio:

Qui abbiamo due item della listview che fa da lista eventi / trasmettitore , essi stessi vengono poi inseriti all’interno all’interno del file di log con data e ora dell’avvenimento.



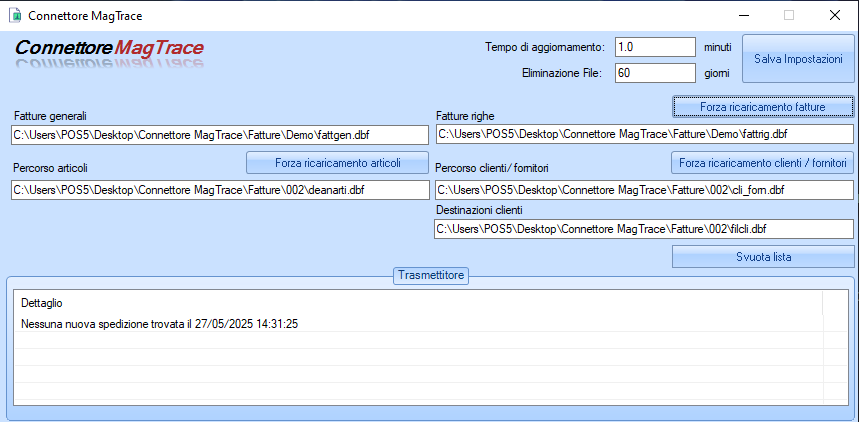
I file di log si trovano all’interno della cartella Log che si trova nel percorso dell’eseguibile , all’interno di essi troveremo tutti i log creati .

N.B. i file di Log e Backup vengono eliminati automaticamente se superano i giorni da noi decisi per l’eliminazione del file.

**Ogni textbox contiene i percorsi dei file DBF**

**Impostazioni sulle variabili di tempo**

## Come si presenta il connettore



**Lista eventi che ci dice cosa sta accadendo in background**

Ci sono dei **pulsanti** per forzare i ricaricamenti, vanno ad eseguire le operazioni

anche se i file DBF non sono stati modificati.