

ETHEA InstantTranslation Tools 7.4 per Applicazioni Delphi e InstantSolutions Framework

Ethea InstantTranslation Tools è serie di strumenti per il supporto alla traduzione in lingua straniera di applicazioni Delphi, sviluppate con Delphi (versioni Delphi 10.4, Delphi 11 e Delphi 12).
Questo documento si riferisce alla versione 7.4.0 – documento aggiornato il 14/11/2023

Indice

Premessa.....	1
L'engine per le traduzioni e la direttiva CBLIB_ITA.....	2
Componenti e tools di traduzione: Translation Manager e Repository Editor.....	2
1. Abilitazione di un progetto Delphi alla gestione multilingua.....	2
2. Inizializzazione ambiente e lingua.....	2
3. Meccanismo di cambio lingua della GUI da parte dell'utente.....	3
4. Abilitazione alla traduzione delle forms, frames, datamodules.....	4
5. Abilitazione alla traduzione dei messaggi.....	4
6. Abilitazione alla utilizzo della form di traduzione "embedded".....	5
7. Creazione/Aggiornamento dei files di traduzione.....	5
8. Locking dei files.....	6
Uso del Translation Manager.....	7
1. Pagina Ambiente.....	7
2. Pagina Spazio di lavoro (riepilogo).....	8
3. Preparazione all'uso con Translation Repository Editor.....	9
4. Pagina Spazio di lavoro (traduzione).....	10
5. Menu Dati.....	11
6. Ricerca.....	11
7. Imposta Filtro.....	11
8. Salvataggio dei dati.....	11
9. Pagina dizionario.....	12
Uso del Repository Editor.....	13
1. Pagina Ambiente.....	13
2. Pagina Dizionario.....	14
3. Traduzione automatica con Microsoft Translate Service (Azure).....	14
4. Revisione delle traduzioni automatiche.....	15
5. Traduzione manuale dei messaggi.....	15
Uso della Form "Embedded" di traduzione.....	16
6. Selezione file da Tradurre.....	16
7. Navigazione sulle diciture da tradurre.....	16
8. Traduzione e aggiornamento finestra.....	16
9. Modalità "PickObject".....	17
10. Selezione e traduzione di un elemento grafico della finestra.....	17
11. Selezione e traduzione di un campo di una tabella.....	17
12. Importazione ed esportazione nel Repository.....	17
Integrazione con Ethea InstantSolutions.....	18
1) Gestione in lingua dei dati:.....	18
2) Gestione in lingua della GUI (interfaccia utente):.....	18

Premessa

Qualsiasi applicazione può usufruire di questo package, a patto di considerare questi punti:Pm

- La lingua di base dell'applicazione da tradurre può essere una lingua qualsiasi (stringhe di messaggi, stringhe nelle forms e frames, datamodules)
- Si vogliono tradurre esclusivamente "stringhe" e non anche immagini, icone o altre parti grafiche.

N.B. dalla versione 7.3.0 è cambiata la struttura dei file xml di traduzione: i campi "OriginalStr" e "TranslationStr" sono passati da 30 caratteri a 250: questo non dovrebbe avere un impatto sui repository già esistenti!

Inoltre i 2 tool forniti (Translation Editor e Repository Editor) sono compilati utilizzando "Skia4Delphi" per le dialog animate.

Inoltre sono stati compilati con l'ultima versione di Delphi disponibile.

L'engine per le traduzioni e la direttiva CBLIB_ITA

Per utilizzare i tools di traduzione all'interno di un progetto InstantSolutions* non è necessario nulla, in quanto le units sono già contenute.

Se non si sta utilizzando InstantSolutions allora è necessario avere delle units che fanno parte della libreria di Ethea e che vengono installate nella cartella src\Engine:

```
src\Engine\CBMultiLanguage.pas
src\Engine\CBMultiLanguageMessages.pas
src\Engine\CBMultiLanguageUtils.pas
src\Engine\CBMultiLanguageVCL.pas
```

Queste units funzionano in 2 modalità: senza alcuna direttiva di compilazione la lingua "base" del sistema di traduzione sarà l'inglese, mentre impostando la direttiva di compilazione CBLIB_ITA (che è il default delle applicazioni InstantSolutions) si imposta la lingua di base "italiana". Questo significa che tale direttiva andrà impostata a livello di progetto e dipenderà dalla lingua utilizzata per sviluppare le form e i messaggi del progetto: se le form e i messaggi sono in italiano impostare CBLIB_ITA tra le direttive di compilazione del progetto.

**InstantSolutions è un prodotto di Ethea per lo sviluppo rapido di applicazioni.*

Componenti e tools di traduzione: Translation Manager e Repository Editor

Uno dei vantaggi è quello di poter tradurre una applicazione "al volo" senza ricompilare: basta includere alcune form (Fbase.pas/dfm, FtrxEmbeddedForm.pas/dfm, FsearchText.pas/dfm, FsetFilter.pas/dfm) nel programma che vuole essere tradotto "al volo". L'"Embedded Form Editor" è form per la traduzione integrata direttamente nell'applicazione.

Inoltre ci sono 2 tools esterni:

1. Il "Translation Manager" per tradurre le finestre e i messaggi dell'applicazione.
2. Il "Repository Editor" per collezionare in un dizionario tutte le "diciture" da tradurre.

1. Abilitazione di un progetto Delphi alla gestione multilingua

Per abilitare una applicazione alla gestione multilingua occorre seguire questi passaggi:

2. Inizializzazione ambiente e lingua

Indicare come già detto la direttiva di compilazione CBLIB_ITA se il proprio progetto è scritto in italiano. Aggiungere ai search-path del progetto la cartella contenente le units CBMultiLanguageXXX della libreria CBLib.

Includere nel file di progetto (dpr) prima di qualsiasi form o datamodule che si vuole tradurre la chiamata ad una procedura (es.RegisterTrxProperties) che "attiva" il multilingua:

es:

```
Application.Initialize;
RegisterTrxProperties;
Application.CreateForm(TFmSimple, fmSimple);
Application.Run;
```

Nella procedura RegisterTrxProperties occorre:

- Richiamare **InitTrxSupport** Indicare la cartella contenente il "dizionario" dei files delle traduzioni (in formato xml) la lingua di traduzione e se l'applicazione deve "aggiornare" questi files:

```
InitTrxSupport(
  ExtractFilePath(Application.ExeName)+'TrxRepository', //Path dizionario
  LUserLanguage, //lingua di 'output' della traduzione (scelta dall'utente)
  {$IFDEF UPDATEREPOSITORY}True,{$ELSE}False,{$ENDIF} //aggiorna i files
  mlItalian //lingua di base dell'applicazione
);
```

- Nel caso si sia indicato True per aggiornare i files, occorre anche indicare quali proprietà dei componenti si vogliono tradurre richiamando **RegisterTrxProperty**:

```
{$IFDEF UPDATEREPOSITORY}
RegisterTrxProperty('TComponent','EditCaption', txString);
RegisterTrxProperty('TComponent','Caption', txString);
RegisterTrxProperty('TComponent','Text', txString);
RegisterTrxProperty('TComponent','Hint', txString);
```

```

RegisterTrxProperty('TField', 'DisplayLabel', txString);
RegisterTrxProperty('TRadioGroup', 'Items', txStrings);
RegisterTrxProperty('TListBox', 'Items', txStrings);
RegisterTrxProperty('TMemo', 'Lines', txStrings);
RegisterTrxProperty('TComboBox', 'Items', txStrings);
RegisterTrxProperty('TCheckBox', 'Items', txStrings);
RegisterTrxProperty('TValueListEditor', 'TitleCaptions', txStrings);
RegisterTrxProperty('TValueListEditor', 'Strings', txStrings);
RegisterTrxProperty('TLabelEdit', 'EditLabel', txBoundLabel);
RegisterTrxProperty('TTabControl', 'Tabs', txStrings);
RegisterTrxProperty('TRichEdit', 'Lines', txStrings);
RegisterTrxProperty('TListView', 'Columns', txListView);
RegisterTrxProperty('TStatusBar', 'Panels', txStatusPanel);
RegisterTrxProperty('TDbGrid', 'Columns', txDbGrid);
RegisterTrxProperty('TDbNavigator', 'Hints', txStrings);
{$ENDIF}

```

- In ogni caso occorre registrare le classi utilizzare per la traduzione:

```

//Tutte le classi passate alla RegisterTrxProperty vanno registrate
RegisterClasses([
  TComponent,
  TField,
  TRadioGroup,
  TListBox,
  TMemo,
  TComboBox,
  TCheckBox,
  TValueListEditor,
  TLabelEdit,
  TTabControl,
  TRichEdit,
  TListView,
  TStatusBar,
  TDbGrid,
  TDbNavigator
]);

```

Nel caso di aggiornamento dei files richiamare per ogni unit di messaggi la procedura di “registrazione” degli stessi (spiegata nella sezione successiva):

```

{$IFDEF UPDATEREPOSITORY}
  //Aggiorno il file dei messaggi
  LsErrorMsg.RegistraMessaggi;
{$ENDIF}

```

3. Meccanismo di cambio lingua della GUI da parte dell'utente

La flessibilità del sistema consente di decidere se la lingua dell'applicazione sia predefinita dentro l'applicazione (attraverso ad esempio una direttiva di compilazione) oppure l'utente possa scegliere di cambiare lingua.

Per far questo occorre utilizzare i 2 metodi:

ReadAppLangFromReg serve per leggere dal registro di Windows con quale lingua deve avviarsi l'applicazione di una certa società. Questa chiamata di solito si mette nell'inizializzazione del sistema di traduzione:

```

LuserLanguage := ReadAppLangFromReg(TRXCOMPANY, TRXAPPLICATION);
InitTrxSupport(ExtractFilePath(Application.ExeName)+TRXREPOSPATH,
  LuserLanguage, //lingua scelta dall'utente
  {$IFDEF UPDATEREPOSITORY}True, {$ELSE}False, {$ENDIF}
  mlItalian //lingua di base dell'applicazione
);

```

WriteAppLangToReg serve per reimpostare la lingua (facendola per esempio scegliere all'utente):

```

WriteAppLangToReg(TRXCOMPANY, TRXAPPLICATION, NewLang);

```

Dopo questa chiamata è buona norma avvisare l'utente che per ottenere la nuova GUI occorre riavviare l'applicativo.

Questa tecnica è stata implementata ed utilizzata dal Translation Manager e dal Repository Editor (vedi capitoli

relativi).

4. Abilitazione alla traduzione delle forms, frames, datamodules

Per abilitare la traduzione di forms, frames e datamodules occorre modificare alcuni metodi della classe stessa. Per questo motivo è buona cosa avere una classe di base (TformBase, TdmBase, TframeBase) ove abilitare la traduzione, dalle quali ereditare tutte le classi dell'applicazione.

- In queste unit aggiungere nella uses:

```
uses
  CBMultiLanguage, CBMultiLanguageVCL;
```

- Nell'interface della classe aggiungere la proprietà FormTranslator e definire questi metodi:

```
TfmSimple = class(TForm)
private
  FFormTranslator: TFormTranslatorVCL;
  function GetFormTranslatorVCL: TFormTranslatorVCL;
protected
  procedure ReadState(Reader: TReader); override;
public
  property FormTranslator: TFormTranslatorVCL read GetFormTranslatorVCL;
  constructor Create(AOwner: TComponent); override;
end;
```

- Implementare i metodi in questo modo:

```
constructor TfmSimple.Create(AOwner: TComponent);
begin
  inherited;
  FormTranslator.DoFormTranslation;
end;

function TfmSimple.GetFormTranslator: TFormTranslator;
begin
  if not assigned(FFormTranslator) then
    FFormTranslator := TFormTranslatorVCL.Create(self);
  Result := FFormTranslator;
end;

procedure TfmSimple.ReadState(Reader: TReader);
begin
  //Questo metodo viene lanciato una volta ogni "livello"
  //di forminheritance quindi il nome della form indica il file
  //che deve essere usato per le traduzioni
  inherited;
  FormTranslator.AddForm(Self, Name);
end;
```

5. Abilitazione alla traduzione dei messaggi

Questo aspetto è sicuramente il meno “elegante” e comporta una maggiore modifica dei programmi: invece di utilizzare resourcestrings occorre trasformare tutti i messaggi in funzioni che restituiscono una stringa e che al loro interno richiamano la GetMsgMultiLanguage. Pur non essendo l'approccio più veloce consente di non tralasciare alcun messaggio e di non dover modificare alcuna parte dell'applicazione. Inoltre permette il meccanismo di “registrazione” dei messaggi “al volo”.

- Esempio: se prima bastava fare:

```
resourcestring
APPLICATION_NAME='Sistema Informativo Finanziario';
CONSTRUCTOR_ERRATO='Constructor errato! Usare %s invece di Create';
```

- Ora occorre scrivere nell'interface:

```
function APPLICATION_NAME : string;
function CONSTRUCTOR_ERRATO : string;
```

- e nell'implementation:

```
const
```

```

UNIT_NAME = 'LsErrorMsg';
function APPLICATION_NAME : string;
begin
  Result := GetMsgMultiLanguage( UNIT_NAME, 'APPLICATION_NAME',
    'Sistema Informativo Finanziario' );
end;
function CONSTRUCTOR_ERRATO : string;
begin
  Result := GetMsgMultiLanguage( UNIT_NAME, 'CONSTRUCTOR_ERRATO',
    'Costruttore errato! Usare %s invece di Create' );
end;

```

In ogni unit di messaggi implementare la procedura di registrazione che deve essere richiamata in caso di aggiornamento dei files di traduzione (vedi sopra):

```

procedure RegistraMessaggi;
begin
  StartMessagesRegistration(UNIT_NAME);
  Try
    APPLICATION_NAME ;
    CONSTRUCTOR_ERRATO ;
  Finally
    EndMessagesRegistration(UNIT_NAME);
  End;
end;

```

6. Abilitazione alla utilizzo della form di traduzione “embedded”

Per poter integrare la form di traduzione “embedded” nell'applicazione, che consente all'utente la traduzione senza utilizzare l'editor esterno occorre includerla nel progetto (dpr):

```

FTrxEmbeddedForm in '{lib}\FTrxEmbeddedForm.pas' {fmTrxEmbeddedForm: TFrame},
FBase in '{lib}\FBase.pas' {fmBase},
FSearchText in '{lib}\FSearchText.pas' {TextSearchDialog},
FSetFiler in '{lib}\FSetFiler.pas' {fmSetFilter},

```

Per poter richiamare la form embedded occorre farlo da una form (possibilmente di base) che abbia implementato la gestione del FormTranslator (vedi sopra) e intercettare la pressione di una combinazione di tasti a scelta (es. CTRL-SHIFT-F12). E' necessario anche forzare KeyPreview a True. Tutte queste impostazioni vanno fatte sulla form di base.

```

procedure TMiaFormDiBase.FormKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
{$IFDEF UPDATEREPOSITORY}
  if (Key = VK_F12) and (Shift = [ssShift,ssCtrl]) then
  begin
    ShowTrxForm(Self, username, FFormTranslator, dmForm.FormTranslator, alRight);
  end;
{$ENDIF}
end;

```

la procedura ShowTrxForm è contenuta nella unit FtrxEmbeddedForm, che va aggiunta in uses. Attenzione allo username che va passato per gestire la concorrenza sui files da tradurre (mentre di usa la form embedded, un altro utente potrebbe utilizzare un editor esterno o un'altra form embedded).

Il parametro di allineamento determina la posizione della form a video. E' buona cosa prevedere questa funzionalità sottoposta alla direttiva di compilazione UPDATEREPOSITORY.

7. Creazione/Aggiornamento dei files di traduzione

La creazione dei files delle traduzione avviene a run-time.

Abilitando l'aggiornamento automatico dei files di traduzione (con la direttiva di compilazione UPDATEREPOSITORY) all'avvio dell'applicazione vengono subito creati tutti i files dei messaggi (grazie al richiamo delle procedure RegistraMessaggi).

Per quanto riguarda le forms, datamodules, frames, i files delle traduzioni vengono generati alla creazione degli stessi, quindi se occorre creare tutti i files occorre prevedere una procedura che apra automaticamente tutte le finestre. Questo è necessario perché il meccanismo dell'ereditarietà visuale di una form si scatena solo a run-time in fase di caricamento dei dfm.

Al primo avvio l'applicazione non presenta alcuna traduzione: **si è scelto di lasciare la dicitura in italiano qualora la traduzione non fosse disponibile**. Questa regola vale soprattutto per i rilasci "successivi". **Non è necessario aver provveduto alla traduzione completa dell'applicazione per poterla far girare in un'altra lingua**.

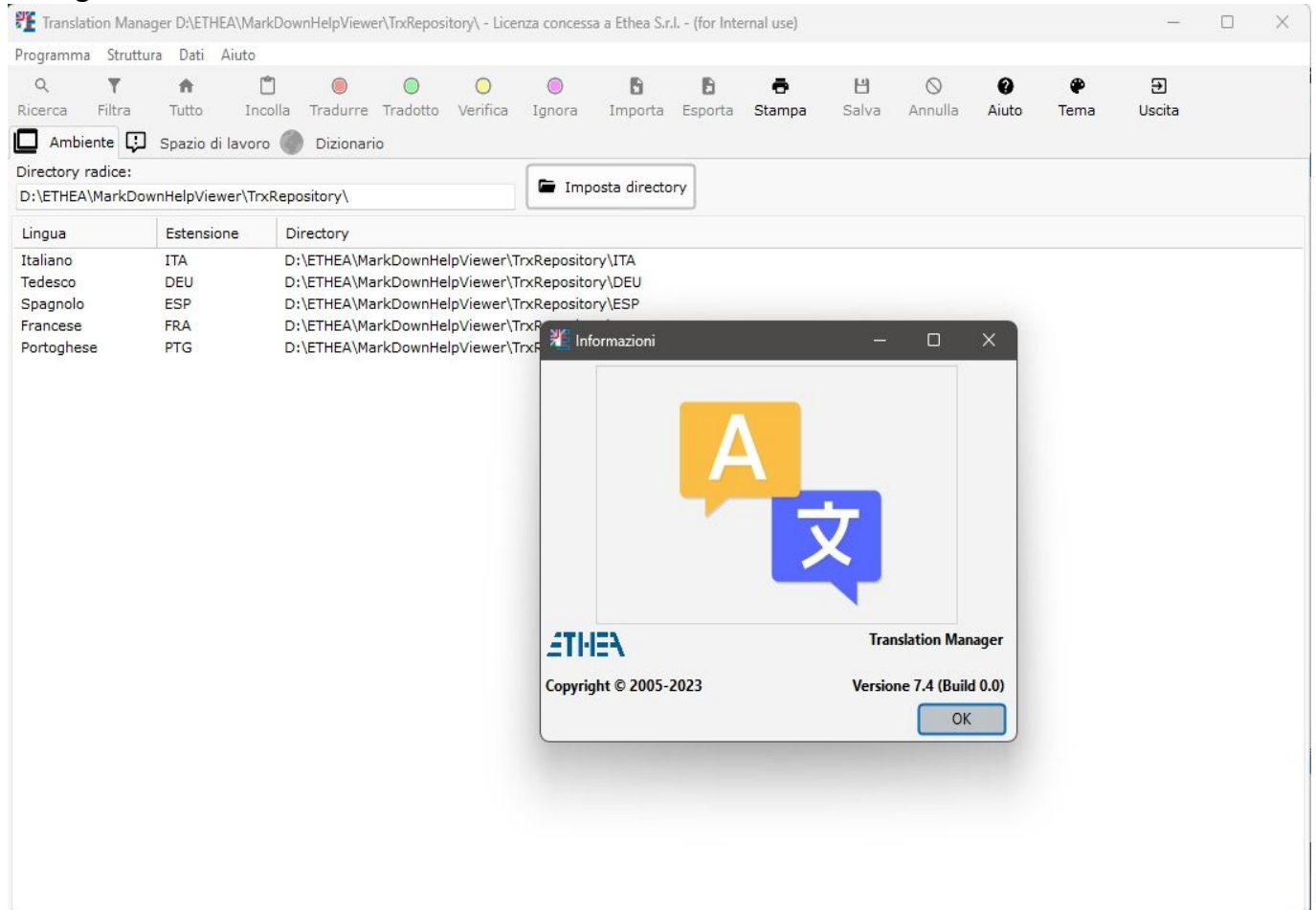
8. Locking dei files

E' stato previsto un meccanismo di locking dei files per prevenire la modifica contemporanea da parte di più utenti degli stessi files per la traduzione. E' un meccanismo semplice basato sulla generazione di un file con estensione **.lck** nel momento in cui un utente entra in "editing". Se per qualsiasi motivo l'applicazione dell'utente dovesse terminare in modo anomalo, lasciando il file sul disco, basta cancellarlo manualmente per ripristinarne l'utilizzo.

Uso del Translation Manager

Il Translation Manager è l'editor per aggiornare i messaggi in lingua. Questo editor lavora sui files XML generati dall'applicazione per la traduzione.

1. Pagina Ambiente

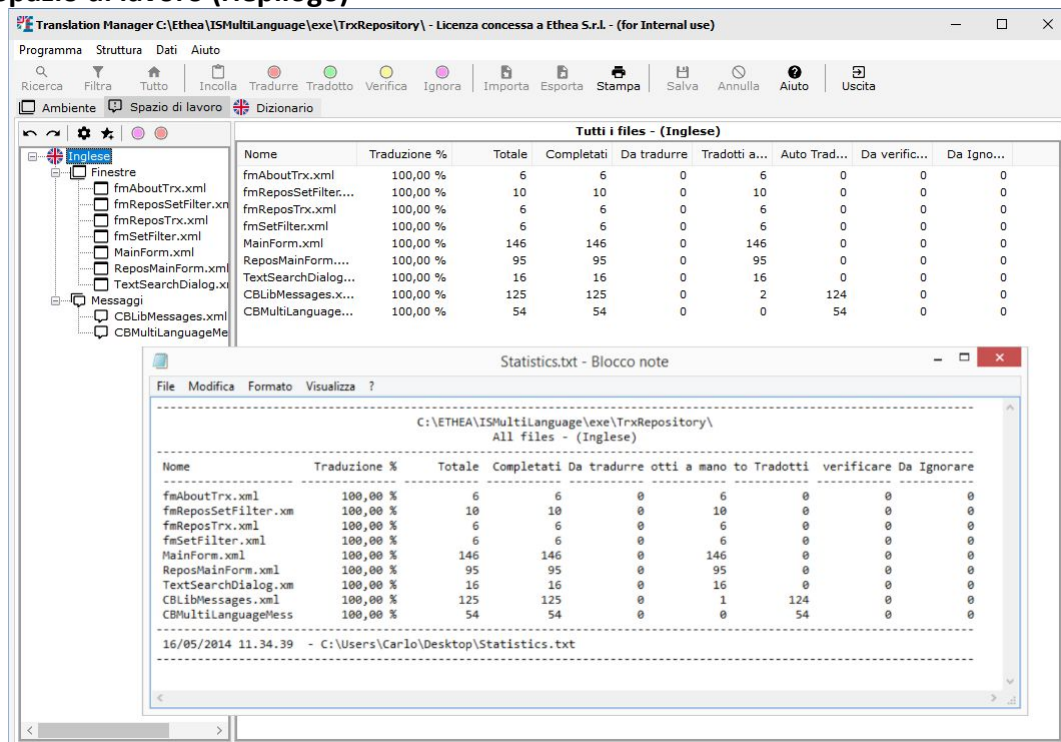


E' necessario conoscere l'ubicazione (cartella) di tali files per poter effettuare la traduzione: nella pagina "Ambiente" occorre indicare la cartella dove c'è il "dizionario" dei files delle traduzioni: è la path specificata in **InitTrxSupport**. La pagina mostra l'elenco delle lingue da tradurre indicandone le sottocartelle.

Per verificare la versione del programma usare nel menu aiuto la voce "informazioni": compare la finestra che riporta la versione del programma.

Il programma stesso è multilingua: italiano/inglese. E' sufficiente selezionare dal menu Programma/Cambia lingua per avere l'interfaccia utente in inglese e viceversa.

2. Pagina Spazio di lavoro (riepilogo)



The screenshot shows the Translation Manager application window. The main area displays a table titled "Tutti i files - (Inglese)" with the following data:

Nome	Traduzione %	Totale	Completati	Da tradurre	Tradotti a...	Auto Trad...	Da verific...	Da Igno...
fmAboutTrx.xml	100,00 %	6	6	0	6	0	0	0
fmReposSetFilter....	100,00 %	10	10	0	10	0	0	0
fmReposTrx.xml	100,00 %	6	6	0	6	0	0	0
fmSetFilter.xml	100,00 %	6	6	0	6	0	0	0
MainForm.xml	100,00 %	146	146	0	146	0	0	0
ReposMainForm....	100,00 %	95	95	0	95	0	0	0
TextSearchDialog...	100,00 %	16	16	0	16	0	0	0
CBLibMessages.x...	100,00 %	125	125	0	2	124	0	0
CBMultiLanguage...	100,00 %	54	54	0	0	54	0	0

The 'Statistics.txt' window shows a printed version of this table with the following header:

```

-----
C:\ETHEA\ISMultiLanguage\exe\TrxRepository\
All files - (Inglese)
-----
Nome          Traduzione %  Totale  Completati  Da tradurre  otti a mano  Tradotti  verificare  Da Ignorare
-----
fmAboutTrx.xml  100,00 %      6         6           0             6           0         0           0
fmReposSetFilter.xml  100,00 %     10        10           0            10          0         0           0
fmReposTrx.xml   100,00 %      6         6           0             6           0         0           0
fmSetFilter.xml  100,00 %      6         6           0             6           0         0           0
MainForm.xml     100,00 %    146       146          0            146         0         0           0
ReposMainForm.xml 100,00 %     95        95           0            95          0         0           0
TextSearchDialog.xml 100,00 %    16        16           0             16          0         0           0
CBLibMessages.xml 100,00 %   125       125          0             1          124        0           0
CBMultiLanguageMess 100,00 %    54        54           0             0           54         0           0
-----
16/05/2014 11.34.39 - C:\Users\Carlo\Desktop\Statistics.txt
-----

```

Lo spazio di lavoro mostra sulla sinistra l'albero contenente tutte le lingue, e per ciascuna lingua l'elenco delle "mappe" (o datamodules) e dei messaggi.

Sulla destra sono visibili due tipi di informazioni.

Se sull'albero si è posizionati su una lingua è visibile il riepilogo delle percentuali di traduzione della lingua selezionata: per ogni mappa o messaggio viene mostrato:

Nome del file (in formato XML UTF-8)

Percentuale della traduzione

Totale diciture completate (tradotte o ignorate)

Totale diciture da tradurre

Totale diciture tradotte (di cui a mano o automaticamente da dizionario)

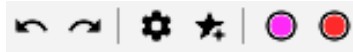
Totale diciture da verificare (in caso di variazione della dicitura in italiano).

Totale diciture da ignorare

Questo riepilogo viene mantenuto costantemente aggiornato ogni volta che si opera una traduzione. E' possibile creare un file di testo (come quello mostrato in figura) di questo riepilogo tramite la funzione di stampa.

3. Preparazione all'uso con Translation Repository Editor

In modalità “riepilogo” è possibile effettuare una serie di operazioni in base alla voce dell'albero selezionata, disponibili sulla toolbar posta sopra l'albero:



E' possibile ridurre o espandere l'albero delle voci.

“Generazione/aggiornamento del dizionario da tradurre: cliccando sul simbolo “ingranaggio” è possibile esportare nel file del dizionario della lingua selezionata tutti i messaggi **compresi quelli non tradotti ma senza includere i messaggi tutti maiuscolo (che si suppone siano nomi di field)**. Se i files del dizionario non dovessero esistere vengono creati automaticamente.

In questa fase si aggiorna anche il “contatore” del dizionario delle occorrenze di una dicitura “doppia”.

Aggiornamento di tutte le diciture di tutte le mappe e i messaggi dal dizionario: cliccando sul simbolo “stella” è possibile effettuare l'operazione inversa: recuperare tutte le traduzione effettuate nel dizionario all'interno di tutte le forms e i messaggi.

Impostazione di tutti i messaggi non tradotti da ignorare: cliccando sul simbolo “quadrato viola” è possibile impostare lo status di “ignora” su tutte le diciture non tradotte del file attualmente selezionato.

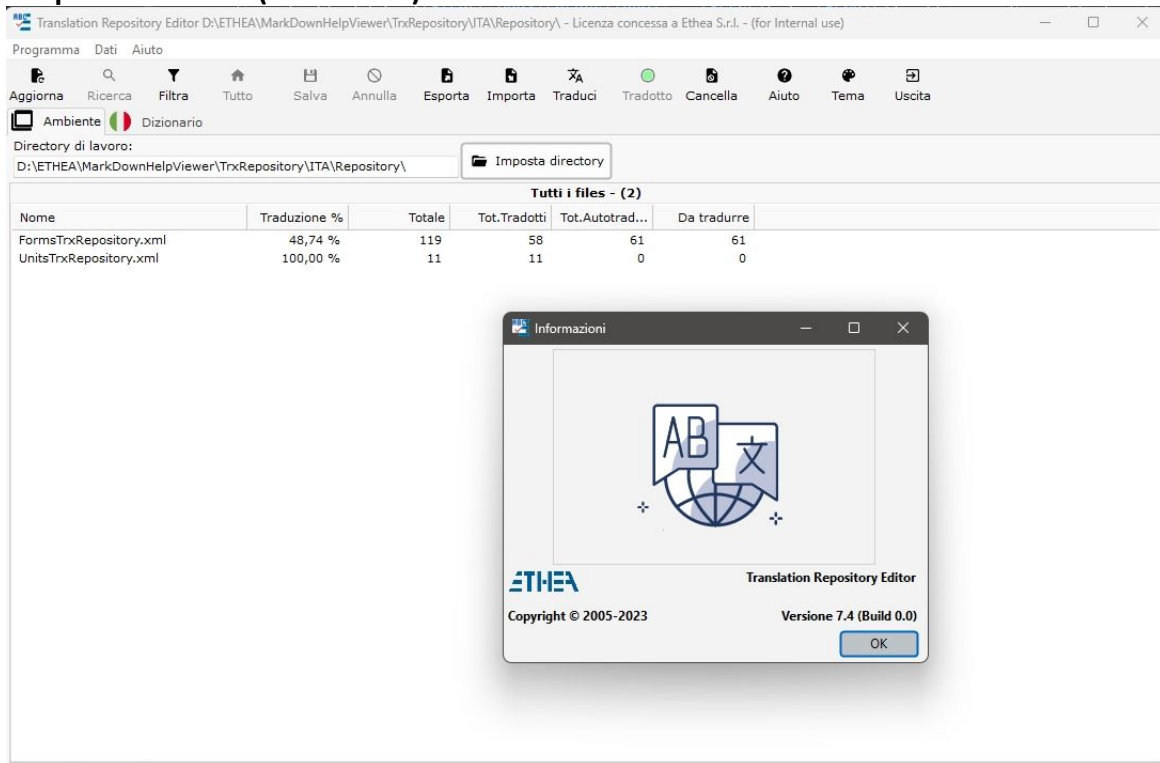
Impostazione di tutti i messaggi da ignorare in “da tradurre”: cliccando sul simbolo “quadrato rosso” è possibile impostare lo status di “da tradurre” su tutte le diciture da ignorare del file attualmente selezionato.

N.B. I file del dizionario per ogni lingua sono 2:

diciture di mappe/frames/datamodules (FormsTrxRepository.xml)

messaggi nelle units (UnitsTrxRepository.xml)

4. Pagina Spazio di lavoro (traduzione)



Selezionando sull'albero di sinistra un file da tradurre, sulla destra appare la lista dei messaggi da tradurre. Nella lista sono indicati:

Icona dello status della traduzione:

- Messaggio da tradurre
- Messaggio tradotto (a mano)
- Messaggio auto-tradotto (tramite dizionario)
- Messaggio da verificare (già tradotto ma è cambiata la dicitura in italiano)
- Messaggio da ignorare

Nome della classe del componente

Nome del componente

Nome della proprietà

Dicitura in italiano

Traduzione in lingua (quella selezionata sull'albero)

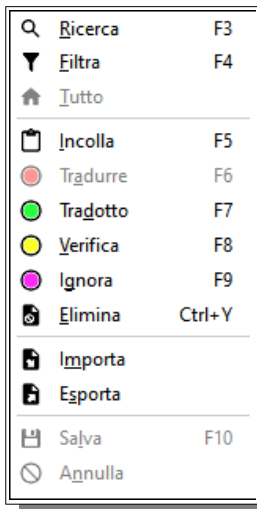
Per tradurre un messaggio occorre posizionare il cursore nello spazio in basso a destra (evidenziato in giallo) e scrivere la traduzione in lingua.

Premendo Invio l'editor passa automaticamente al primo messaggio da tradurre.

Qualora il messaggio in italiano sia su più righe occorre inserire le stesse righe anche nella traduzione. In questo caso premendo Invio l'editor va a capo all'interno del box; solo dopo aver inserito tutte le righe si passa al messaggio successivo.

E' possibile ordinare la lista cliccando sul titolo della colonna corrispondente: la colonna viene evidenziata in grassetto (vedi figura).

5. Menu Dati



Nel file di traduzione è disponibile il menù dei dati con una serie di funzioni:

F3 – Avvia la ricerca (vedi sotto)

F4 – Imposta un filtro (vedi sotto)

Mostra tutto (azzerà il filtro impostato)

F5 – Incolla: copia le diciture già tradotte nelle stesse ancora da tradurre

F6 – Imposta lo status “da tradurre”

F7 – Imposta lo status in “tradotto”

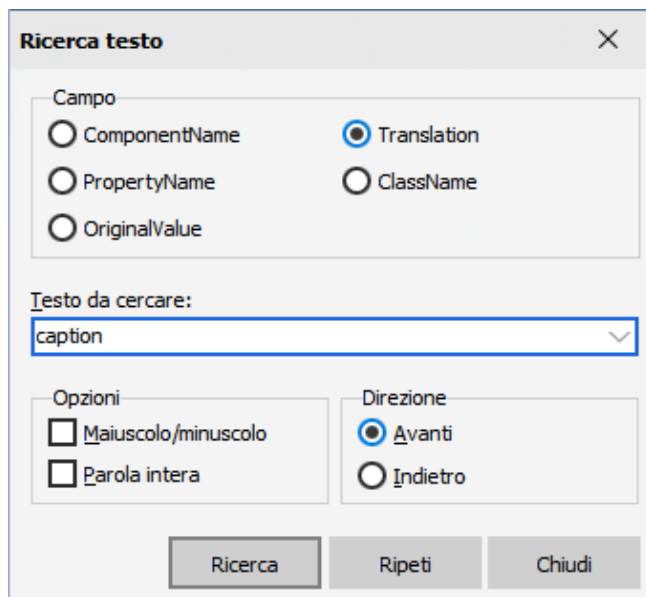
F8 – Imposta lo status “da verificare”

F9 – Imposta lo status “da ignorare”

Alcune di queste funzioni sono disabilitate in base al contesto (es. non si abilita lo status “da tradurre” per le diciture già tradotte, ma occorre “svuotare” a mano la traduzione.

Importa da dizionario: consente di tradurre “automaticamente” le diciture del file corrente prendendole dal dizionario

Esporta in dizionario: esporta le traduzioni del file corrente nel dizionario comprese le diciture tutte maiuscole se sono state tradotte.



6. Ricerca

Nella finestra di ricerca è possibile cercare un testo contenuto in una delle colonne della lista premendo “ricerca”.

E' possibile ripetere la ricerca più volte senza uscire dalla finestra di ricerca premendo “ripeti”.

Le opzioni di ricerca e la ricerca condizionano la ricerca del testo indicato.

Una volta “raggiunto” il messaggio che si intendeva ricercare basta chiudere la finestra.

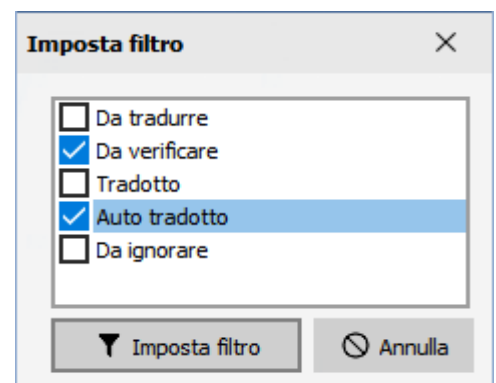
Se la ricerca non ha esito nel file corrente è possibile proseguirla nei restanti files.

Filtro

Nella finestra di filtro è possibile impostare la visualizzazione dei messaggi in base al loro status.

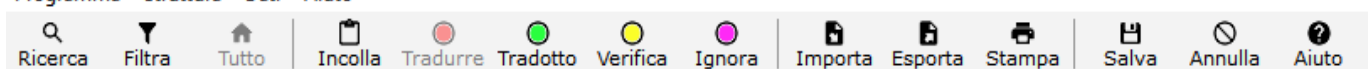
E' molto utile impostare il filtro solo sui messaggi da tradurre. Man mano che verranno tradotti “spariranno” dalla lista.

7. Imposta



8. Salvataggio dei dati

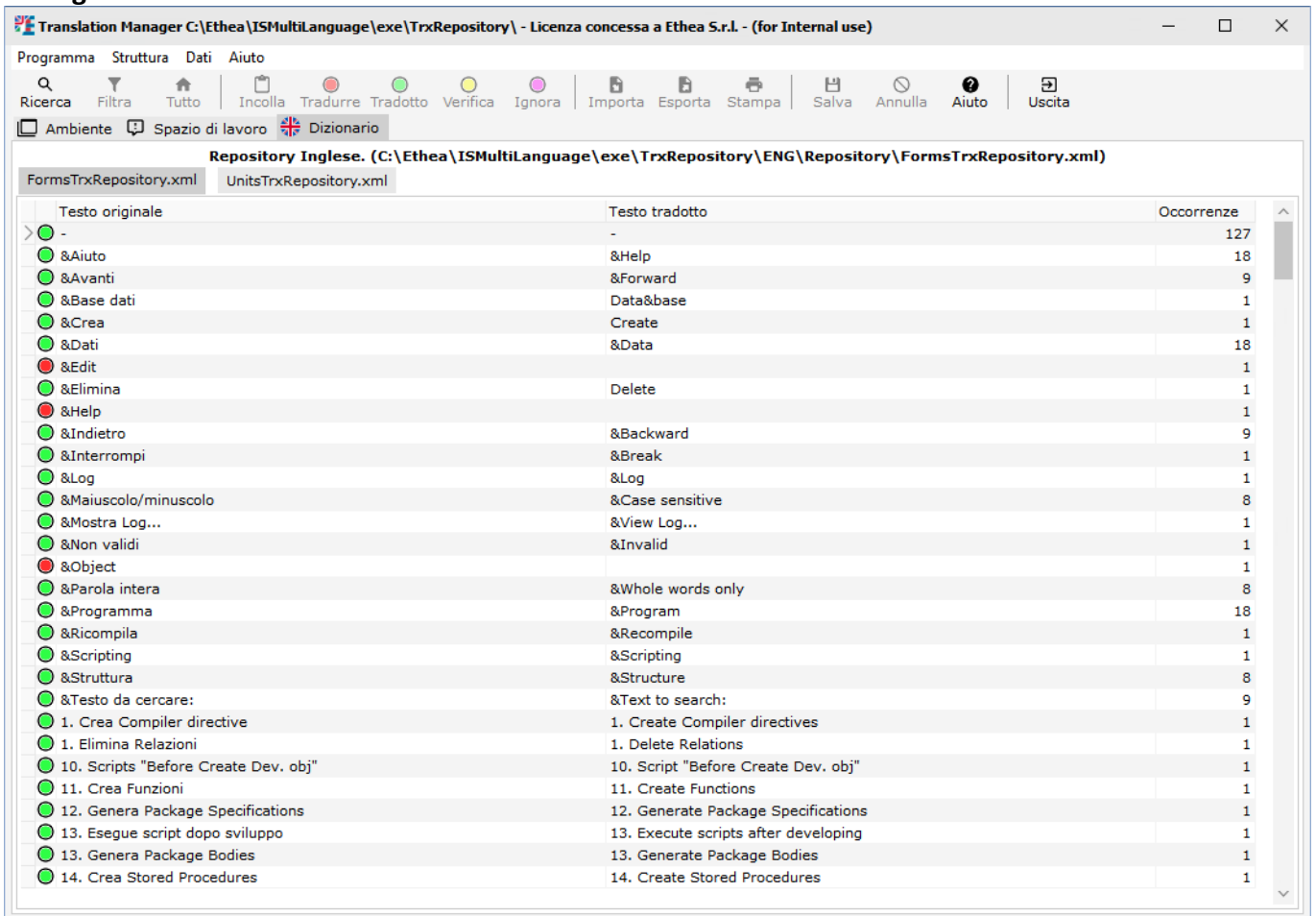
Programma Struttura Dati Aiuto



Tutte le funzioni principali sono disponibili sulla toolbar principale.

Ogni volta che si modifica un file di traduzioni si attivano i pulsanti di salvataggio (F10) o di abbandono (vedi figura).

9. Pagina dizionario



Testo originale	Testo tradotto	Occorrenze
-	-	127
&Aiuto	&Help	18
&Avanti	&Forward	9
&Base dati	Data&base	1
&Crea	Create	1
&Dati	&Data	18
&Edit		1
&Elimina	Delete	1
&Help		1
&Indietro	&Backward	9
&Interrompi	&Break	1
&Log	&Log	1
&Maiuscolo/minuscolo	&Case sensitive	8
&Mostra Log...	&View Log...	1
&Non validi	&Invalid	1
&Object		1
&Parola intera	&Whole words only	8
&Programma	&Program	18
&Ricompila	&Recompile	1
&Scripting	&Scripting	1
&Struttura	&Structure	8
&Testo da cercare:	&Text to search:	9
1. Crea Compiler directives	1. Create Compiler directives	1
1. Elimina Relazioni	1. Delete Relations	1
10. Scripts "Before Create Dev. obj"	10. Script "Before Create Dev. obj"	1
11. Crea Funzioni	11. Create Functions	1
12. Genera Package Specifications	12. Generate Package Specifications	1
13. Esegue script dopo sviluppo	13. Execute scripts after developing	1
13. Genera Package Bodies	13. Generate Package Bodies	1
14. Crea Stored Procedures	14. Create Stored Procedures	1

Nella pagina Dizionario è possibile visualizzare i dizionari di tutte le lingue dell'applicazione ed effettuare la ricerca all'interno dei dizionario stesso.

Come già spiegato è possibile autotradurre le diciture a partire dal dizionario. I messaggi tradotti in questo modo vengono "marcati" con lo status "auto-tradotto" perché non sempre la traduzione letterale è corretta nel contesto in cui si utilizza.

Sia il Translation Manager che il Repository Editor sono applicazione Delphi tradotte in inglese con questi stessi translation tools.

E' possibile ordinare il dizionario cliccando sul titolo della colonna all'interno della griglia.

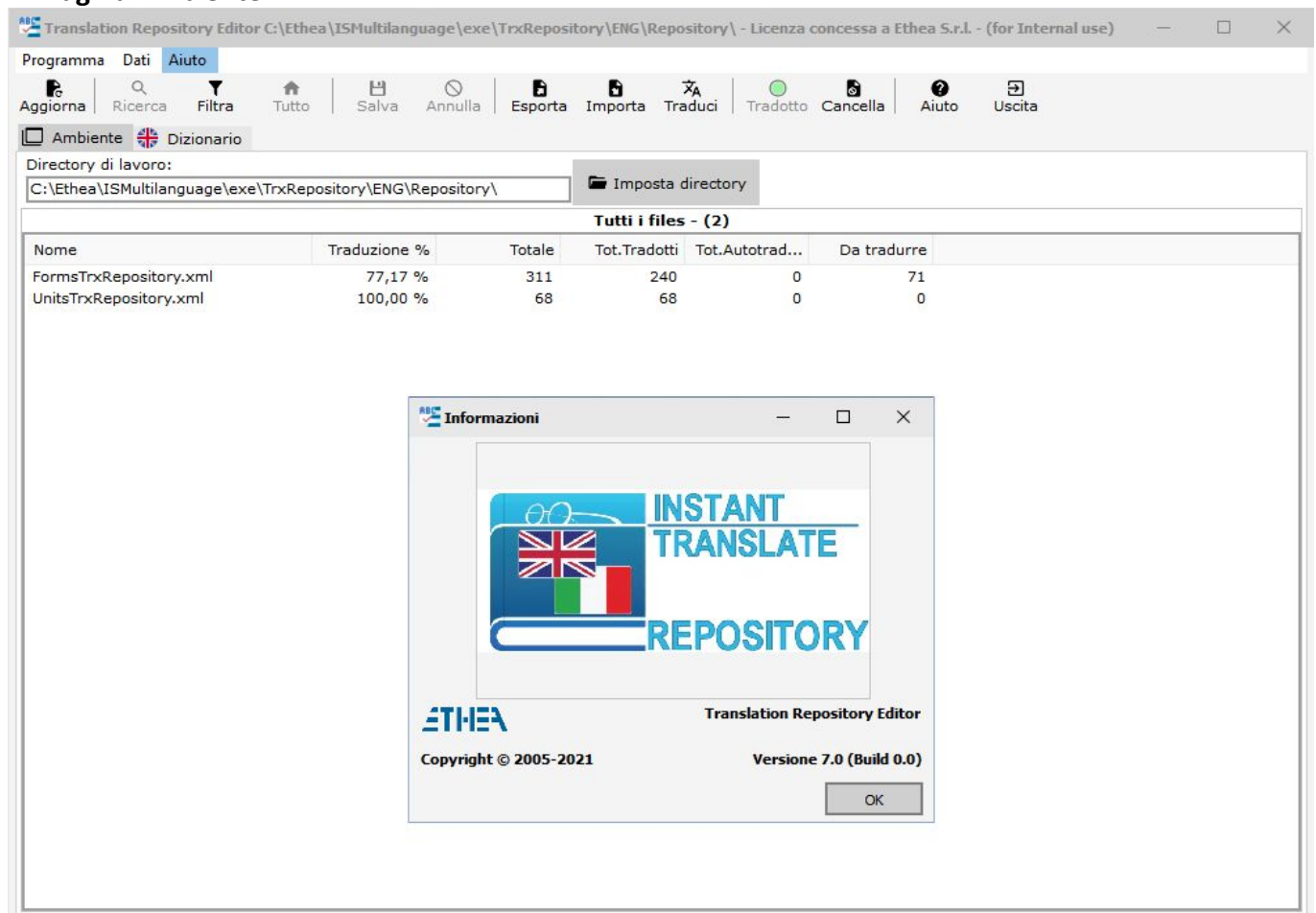
Uso del Repository Editor

E' possibile utilizzare il Repository Editor per tradurre tutti i messaggi dell'applicazione precedentemente esportati.

Per poter suddividere il lavoro è possibile anche “dividere” il file di repository in sottofiles (vedi impostazione filtro e esportazione file) per poter lavorare in più persone alla fase di traduzione.

Una volta suddiviso il lavoro è possibile reimportare i singoli files nel file di repository e utilizzando la funzione di “importazione multipla” del Translation Editor si aggiornano tutte le finestre e i messaggi dell'applicazione.

1. Pagina Ambiente



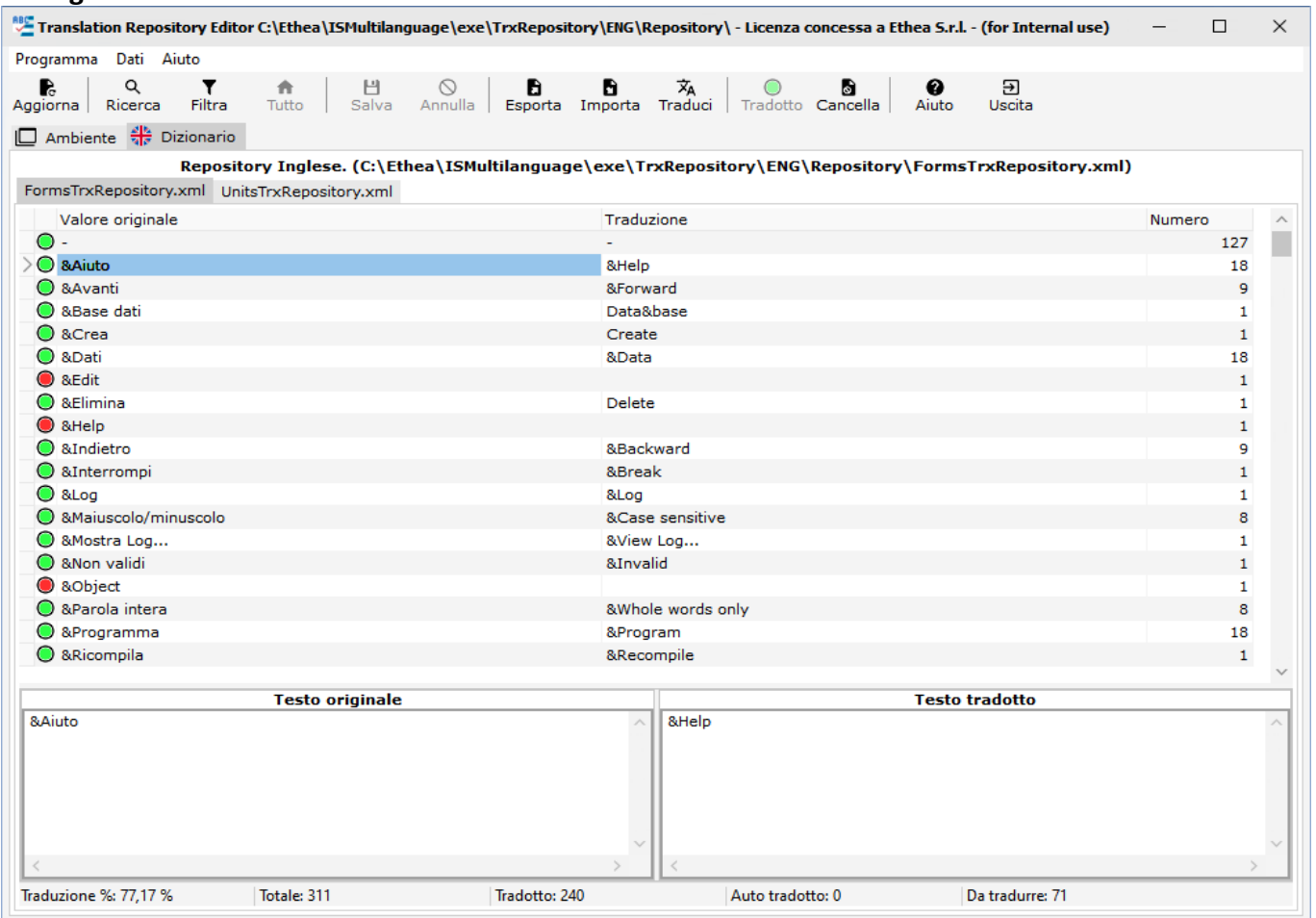
Nella pagina “Ambiente” occorre indicare la cartella dove c'è il “dizionario dati” di una singola lingua.

La pagina mostra l'elenco dei files (che possono essere più di uno qualora il repository principale sia stato spezzato in più sottofiles). A fianco sono riportati i dati statistici: percentuale di traduzione, totale messaggi, totale messaggi tradotti e totale messaggi da tradurre.

Per verificare la versione del programma usare nel menu aiuto la voce “informazioni”: compare la finestra che riporta la versione del programma.

Il programma stesso è multilingua: italiano/inglese. E' sufficiente selezionare dal menu Programma/Cambia lingua per avere l'interfaccia utente in inglese e viceversa.

2. Pagina Dizionario



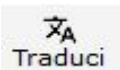
Nella parte bassa della finestra sono visibili i dati “statistici” del file corrente.

3. Traduzione automatica con Microsoft Translate Service (Azure)

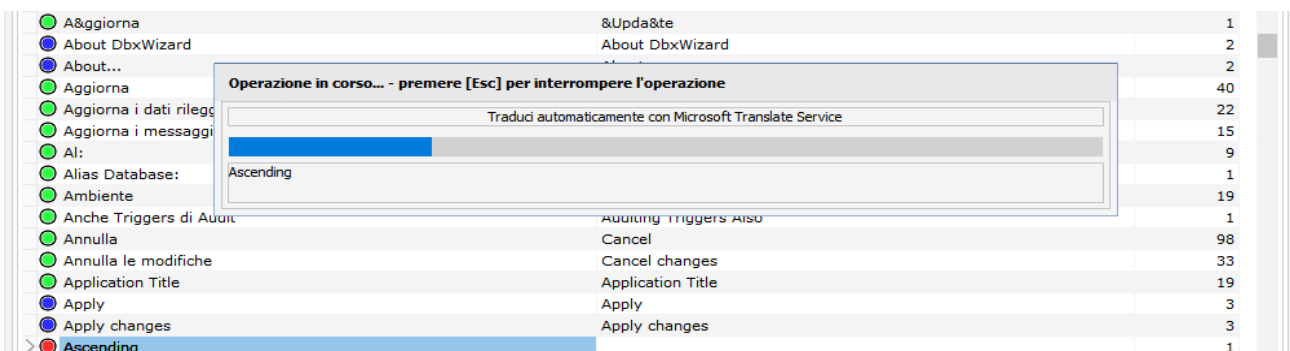
Con InstantTranslation Tools è possibile automatizzare il processo di traduzione sfruttando i servizi Microsoft di traduzione. Per abilitare tale funzione è necessario acquistare tramite il proprio account Microsoft i servizi di traduzione ed ottenere una SubscriptionKey.

Creare quindi nella stessa cartella dell’eseguibile il file MSttranslation.ini:

```
[AzureAccount]
SubscriptionKey=xxxx
[Language]
BaseLanguage=it
```



A questo punto il pulsante “Traduci” si abiliterà, consentendo la traduzione automatica di tutti i messaggi non ancora tradotti (quelli già tradotti vengono saltati).



Al termine del processo di traduzione tutte le traduzioni effettuate in modo automatico vengono marcate con lo

status **“autotradotto” (colore blu)**, in modo tale da evidenziare che devono essere verificate.

4. Revisione delle traduzioni automatiche

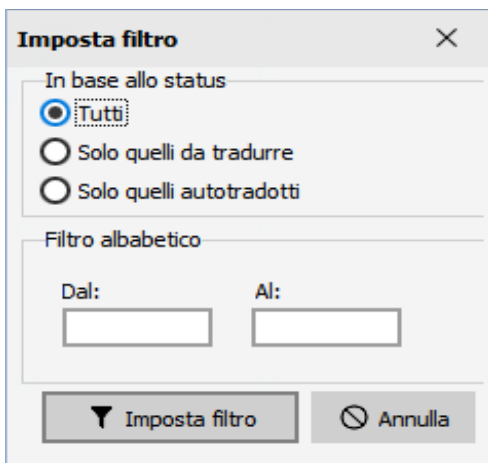
E' importante sapere che i messaggi **“AutoTradotti”** non vengono pescati dal TranslationEditor, quindi occorre impostare il filtro sui messaggi **“autotradotti”** e procedere alla loro revisione/validazione. Se la traduzione è corretta è sufficiente premere sul pulsante **“Tradotto”** che imposta il semaforo verde, altrimenti basta editare la traduzione come si fa normalmente per la traduzione manuale dei messaggi (vedi capitolo successivo).

5. Traduzione manuale dei messaggi

Nella pagina dizionario è possibile tradurre i messaggi contenuti posizionando il cursore della zona in basso a destra evidenziata in giallo. Il semaforo rosso indica i messaggi ancora da tradurre. E' anche possibile cancellare records (es. i record riportanti i nomi dei campi specificati in italiano. L'ultima colonna mostra le occorrenze delle diciture identiche.

Qualora la dicitura del messaggio originale debba essere **“copiata”**, perché identica anche nella lingua di traduzione, basta premere il pulsante **“Tradotto”** con il semaforo verde.

Per poter lavorare in più persone è possibile spezzare il dizionario in sottofiles: prima occorre filtrare i dati indicando per esempio un range da A a C:



Una volta **“filtrato”** il dizionario basta **“esportarlo”** in un altro file chiamandolo ad esempio: TrxReposA_C.xml. Se questo file viene salvato nella stessa cartella del Repository principale verrà mostrato e sarà utilizzabile.

Al termine del lavoro sarà necessario reimportare tutti i singoli files spezzati nei due files **“veri”** del dizionario dal nome FormsTrxRepository.xml e UnitsTrxRepository.xml

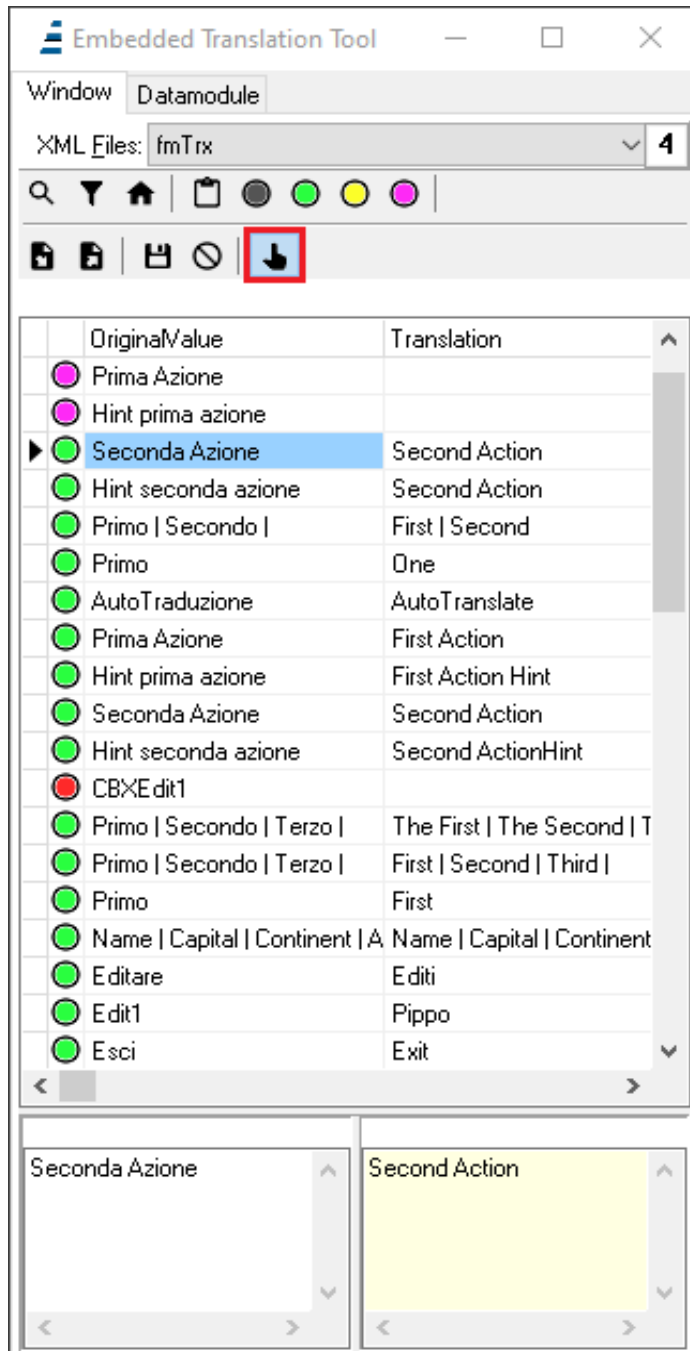
A questo punto occorrerà riutilizzare Translation Manager per aggiornare tutte le finestre e i messaggi dell'applicazione.

E' sempre disponibile la finestra di ricerca con le stesse modalità spiegate per il Translation Manager.

E' possibile ordinare il dizionario cliccando sul titolo della colonna della griglia, oltre che cercare velocemente delle parole digitando le lettere direttamente dentro la griglia.

Uso della Form “Embedded” di traduzione

Una volta preparata l'applicazione con la form “embedded” e decisa quale combinazione di tasti utilizzare, premere tale combinazione (es. CTRL-SHIFT-F12) per visualizzare la finestra “Embedded” di traduzione, che resta “collegata” alla **finestra attiva del programma**.



I simboli utilizzati in questa finestra sono gli stessi utilizzati all'interno dell'editor esterno per le traduzioni, quindi per quanto riguarda il significato e il funzionamento dei tasti contenuti nella toolbar rimandiamo al capitolo corrispondente. Anche la modalità di traduzione tramite la finestra in basso a destra (evidenziata in giallo) rimandiamo al capitolo corrispondente.

Pagine Finestra e Modulo dati

Cambiando la pagina attiva è possibile mostrare gli elementi da tradurre che sono presenti sulla form o nel datamodule della finestra attiva.

6. Selezione file da Tradurre

Dalla combobox “XML Files:” è possibile cambiare il file da tradurre. Sulla destra è mostrato in rosso il numero di files da tradurre, in quanto una finestra attiva del programma può essere frutto di più livelli di “ereditarietà”.

7. Navigazione sulle diciture da tradurre

Spostando il cursore sulle diciture da tradurre, sulla finestra attiva viene evidenziato in rosso l'elemento grafico selezionato. Qualora l'elemento grafico selezionato non sia presente sulla pagina corrente della finestra attiva verrà cambiata automaticamente la pagina per poterlo mostrare.

E' sempre possibile effettuare le ricerche o impostare i filtri su questi elenchi, sempre come avviene nel Translation Manager.

8. Traduzione e aggiornamento finestra

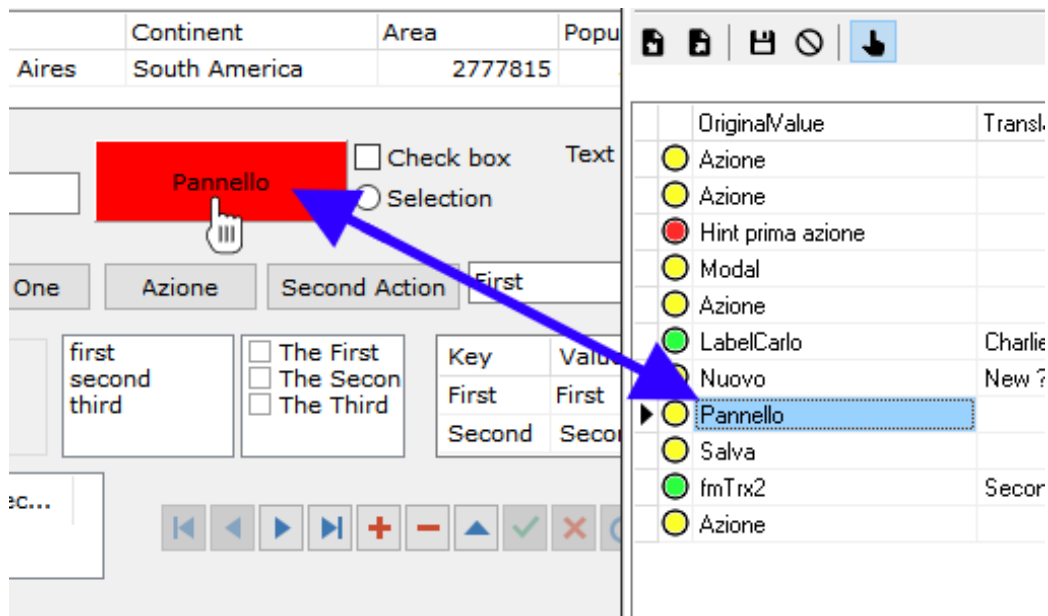
Una volta tradotte le diciture, occorre salvare il file. Una volta salvato il file sulla finestra attiva è già visibile la dicitura tradotta.

9. Modalità “PickObject”

Il modo più semplice per effettuare le traduzioni di una finestra è dato dalla modalità “PickObject” che è attivabile premendo il pulsante con la “manina” evidenziato con il contorno rosso in figura.

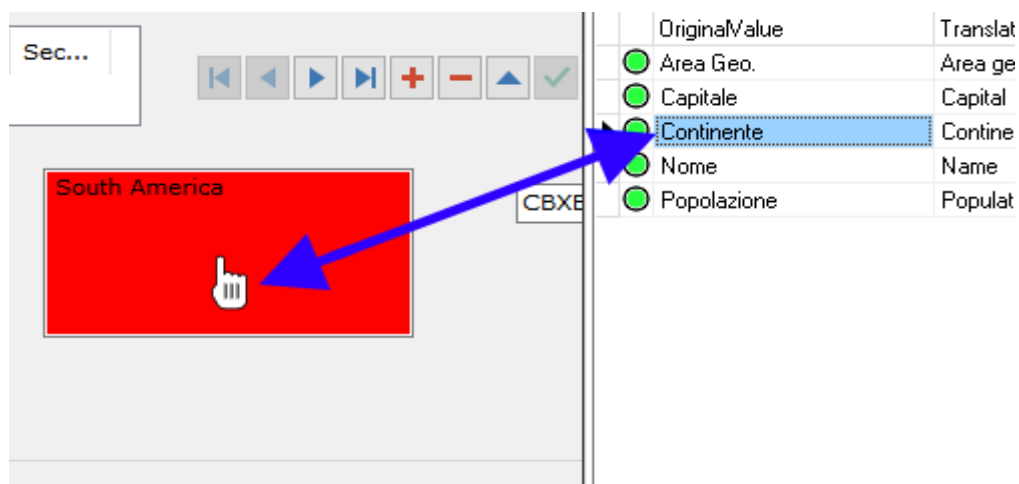
Entrando in questa modalità operativa la finestra attiva viene in qualche modo “addormentata” nelle sue funzionalità classiche, mentre si attiva una modalità di “selezione” dell’oggetto da tradurre. Il cursore si trasforma in una mano e tutti gli elementi visibili sulla finestra attiva diventano “selezionabili” semplicemente cliccandoci sopra.

10. Selezione e traduzione di un elemento grafico della finestra



Se la finestra “embedded” è impostata sulla **pagina “Finestra”**, cliccando con la “manina” sull’oggetto che si vuole tradurre, il “cursore” nella finestra “embedded” di traduzione si sposta automaticamente sulla riga corrispondente per consentire la traduzione dell’elemento cliccato.

11. Selezione e traduzione di un campo di una tabella



Se la finestra “embedded” è impostata sulla **pagina “Modulo Dati”**, cliccando con la “manina” sul campo di edit del campo che si vuole tradurre, il “cursore” nella finestra “embedded” di traduzione si sposta automaticamente sulla riga corrispondente per consentire la traduzione del campo.

12. Importazione ed esportazione nel Repository

Le funzioni di importazione ed esportazione da e verso il Repository sono disponibili anche in questa finestra, così come la funzione di “replica” dei messaggi già tradotti. E' possibile utilizzare indifferentemente la form embedded

o il tool esterno: essi lavorano sugli stessi files utilizzando un meccanismo di locking che previene la traduzione contemporanea della stessa finestra da parte di 2 utenti.

Integrazione con Ethea InstantSolutions

InstantSolutions è il framework di sviluppo in Delphi realizzato da Ethea e basato sulla tecnologia OPF InstantObjects. Tale framework si integra perfettamente con i tools descritti in questo documento, in quanto è stato progettato per contenere già tutti i prerequisiti necessari.

Occorre ricordare che nativamente è possibile sviluppare applicazioni in lingua italiana o inglese con InstantSolutions: questa è la lingua di base, sia per il database che per la GUI dell'applicazione.

Con il supporto multilingua è possibile, all'interno di ISWorkbench, identificare alcuni attributi delle classi come "multilingua": a questi attributi è possibile fornire anche una descrizione in lingua alternativa a quella di default (solo se tra le lingue utilizzate ci sono l'italiano e l'inglese). Questa opzione è necessaria solo se si intende avere una GUI multilingua.

Il concetto di multilingua si riferisce a 2 aspetti complementare ma diversi:

1) Gestione in lingua dei dati:

ISWorkbench si occupa automaticamente di:

a) generare le classi di InstantObjects con un meccanismo automatico di lettura/scrittura degli attributi dal database in base alla lingua attiva: se non trova il dato, utilizza il campo di base (nella lingua di default del database) e lo indica tra parentesi quadre.

b) Aggiornare la struttura del database aggiungendo i campi multilingua (preceduti dal prefisso della lingua es. ITA_DX, ENG_DX).

A livello applicativo non è necessario modificare alcun programma, layout di mappa o report: la lingua che pilota il meccanismo di lettura/scrittura degli attributi è inizializzata in base alla lingua dell'utente, ma è possibile cambiarla al volo, in base al contesto (es. prima di stampare un report in lingua).

2) Gestione in lingua della GUI (interfaccia utente):

Complementare, ma non necessario, è la possibilità di avere una interfaccia utente in lingua.

Se in ISWorkbench si definiscono anche le descrizioni (estesa/mappa/elenco) nella lingua alternativa a quella di base dell'applicativo, ISWorkbench si occupa automaticamente di generare tali descrizioni in modo compatibile con "Ethea Translation Tools". Seguendo poi le linee guida definite nel documento EtheaTranslationTools.pdf e utilizzando il TranslationEditor e il RepositoryEditor sarà possibile creare una interfaccia utente multilingua dell'applicazione.

Per abilitare l'aggiornamento dei file xml necessari al TranslationEditor per le traduzioni di una applicazione ISF, occorre impostare nel file ini dell'applicazione, nella sezione multilingua, il parametro UpdateRepository a Yes e avviare l'applicazione.

All'avvio dell'applicazione parte la procedura che automaticamente aggiornati i files relativi ai messaggi e alle classi dell'applicazione, se nella unit UmultiAppSpecific si implementa il metodo:

```
[MultiLanguage]
LANGUAGES=ITA;ENG;FRA;ESP;DEU
DB_LANGUAGE=ITA
UpdateRepository=Yes
```

N.B. Se si sviluppa con la stessa versione di ISF su 2 progetti diversi, uno multilingua e l'altro no, quando si passa da un progetto all'altro è sempre necessario rigenerare con ISWorkbench le classi, perché le units di Framework sono incompatibili.