unit dm\_TipoAtto;

interface

uses

System.SysUtils, System.Classes, dm\_cnt, Data.DB, Data.FMTBcd,

Datasnap.Provider, Datasnap.DBClient, FireDAC.Phys.FB, FireDAC.Stan.Intf,

FireDAC.Stan.Option, FireDAC.Stan.Param, FireDAC.Stan.Error, FireDAC.DatS,

FireDAC.Phys.Intf, FireDAC.DApt.Intf, FireDAC.Stan.Async, FireDAC.DApt,

FireDAC.Comp.DataSet, FireDAC.Comp.Client, FireDAC.DBX.Migrate, ClientViewFD,

FireDAC.UI.Intf, FireDAC.Stan.Def, FireDAC.Stan.Pool, FireDAC.Phys,

FireDAC.VCLUI.Wait, FDSQLConnection;

type

TdmTipoAtto = class(Tdmcnt)

dsTipoAtto: TDataSource;

cdsTipoAtto: TClientDataSet;

dspTipoAtto: TDataSetProvider;

qxTipoAtto: TFDQuery;

qxMaxProgTipoAtto: TFDQuery;

scvCercaTipoAtto: TFDQView;

cdsTipoAttoPROG\_TIPO\_ATTO: TIntegerField;

cdsTipoAttoDESCR\_TIPO\_ATTO: TStringField;

scvCercaTipoAttoPROG\_TIPO\_ATTO: TIntegerField;

scvCercaTipoAttoDESCR\_TIPO\_ATTO: TStringField;

procedure cdsTipoAttoAfterDelete(DataSet: TDataSet);

procedure cdsTipoAttoAfterPost(DataSet: TDataSet);

procedure cdsTipoAttoBeforePost(DataSet: TDataSet);

procedure cdsTipoAttoReconcileError(DataSet: TCustomClientDataSet;

E: EReconcileError; UpdateKind: TUpdateKind;

var Action: TReconcileAction);

private

{ Private declarations }

public

{ Public declarations }

end;

implementation

uses

cdsutil, fm\_UGrecerror;

{%CLASSGROUP 'Vcl.Controls.TControl'}

{$R \*.dfm}

procedure TdmTipoAtto.cdsTipoAttoAfterDelete(DataSet: TDataSet);

begin

inherited;

CdsApplyUpdates(cdsTipoAtto, 'PROG\_TIPO\_ATTO', bCanRetry);

end;

procedure TdmTipoAtto.cdsTipoAttoAfterPost(DataSet: TDataSet);

begin

inherited;

CdsApplyUpdates(cdsTipoAtto, 'PROG\_TIPO\_ATTO', bCanRetry);

end;

procedure TdmTipoAtto.cdsTipoAttoBeforePost(DataSet: TDataSet);

begin

inherited;

// Controllo campi obbligatori

if((cdsTipoAtto.FieldByName('DESCR\_TIPO\_ATTO').IsNull) or

(cdsTipoAtto.FieldByName('DESCR\_TIPO\_ATTO').AsString = '') )then

raise Exception.Create('Definire la descrizione del tipo atto!');

// Calcolo progressivo chiave

if((cdsTipoAtto.State = dsInsert) and

(cdsTipoAtto.FieldByName('PROG\_TIPO\_ATTO').IsNull)) then

begin

// Calcolo del progressivo

qxMaxProgTipoAtto.Close;

qxMaxProgTipoAtto.Open;

cdsTipoAtto.FieldByName('PROG\_TIPO\_ATTO').AsInteger := qxMaxProgTipoAtto.FieldByName('MAX\_PROG\_TIPO\_ATTO').AsInteger + 1;

end; //if((cdsTipoAtto.State = dsInsert) and ...

if(cdsTipoAtto.FieldByName('PROG\_TIPO\_ATTO').IsNull) then

raise Exception.Create('Definire il progressivo del tipo atto!');

end;

procedure TdmTipoAtto.cdsTipoAttoReconcileError(DataSet: TCustomClientDataSet;

E: EReconcileError; UpdateKind: TUpdateKind; var Action: TReconcileAction);

begin

inherited;

bCanRetry := false;

if AnsiPos('XPKTIPO\_ATTO', E.Message) > 0 then

begin

if (UpdateKind = ukInsert) then

begin

// Calcolo del progressivo

qxMaxProgTipoAtto.Close;

qxMaxProgTipoAtto.Open;

Dataset.FieldByName('PROG\_TIPO\_ATTO').NewValue := qxMaxProgTipoAtto.FieldByName('PROG\_TIPO\_ATTO').AsInteger + 1;

Action := TReconcileAction.raCorrect

end

end

else Action := HandleReconcileError(Dataset, UpdateKind, E);

if Action = TReconcileAction.raCorrect

then bCanRetry := true

end;

end.