1 Disclaimer

This software is delivered as it is. The author assumes no liability for damages, direct or consequential, which may result from it's use.

2 Copyright / Licensing

The software is owned by gig mbh berlin (www.gig-mbh.de).

Two different licenses are available:

1. Free License

   Everyone who wants to use the free license has to register with his full name and address via support@gig-mbh.de.

   Every software where parts of our free software were used for development has to be free also including source code.

   If you derive anything from our software it must be clearly stated that it was derived from it.

   Full source code is included.

2. Extended License

   Licenses have to be bought by a per developer basis. Site licenses would be available on demand.

   Applications built with this software could be deployed without royalty fees. They can be sold and don't need to include source code.

   Distribution of a derived version of our software is only allowed with the explicit agreement of the author.

   Full source code is included.

3 Support

Support is available via email at support@gig-mbh.de for free but it need not remain so in the future.

4 Introduction

This document describes the function of the component TTextFileDataProviderEC.

This class implements the interface defined in TDataProviderEC to synchronise our TMemTableEC dataset component with textfiles of a CSV like format. Every field is seperated by a delimiter you could specify and every record is delimited by <cr> or <cr/lf>. If any special charaters (<cr>, <lf>, “, <delimiter>) are part of the field value itself the value has to be enclosed in quotes (“). Then quotes have to be doubled to be part of a field value. The names of the texfile fields can be read automatically from the first line of the file or can be assigned manually.

For working properly you have to enter field assignments where you specify which filelds in the textfile belongs to which fields in the TMemTableEC component.

The component is completely written in C++ and was developed under C++Builder 5 Pro but it should be usable on C++ Builder 6 if compiled in it’s environment.

Questions, bug reports , enhancement requests, suggestions for improving the docs and comments should be send to support@gig-mbh.de.

5 Properties

Stream

Description: Specifies a TStream descendant where the textdata is contained. If you are working with a normal file you could also set the FileName property instead.

Definition : __property TStream *Stream = {read=FStream, write=SetStream}

Type: public

FileName

Description: Specifies a filename where the data is read from or written to. You could also use the Stream property instead.

Definition : __property AnsiString FileName = {read=FFileName, write=FFileName}

Type: published

FieldAssignments

Description: This property specifies the releation between TMemTableEC and the textfile fields. For every field assignment you specify one line in the following syntax: <memtabfield>;<textfilefield>. If you want to have fields in the TMemTabEC data set which have no direct related field in your textfile but are derived from them in any way you could insert a so called virtual field (simply a field name which does no exist and assign/get the computed values in the data provider's SetMemTabFieldValue and SetTextFileFieldValue event handlers. For more information on the textfile fields see the FieldNames property.

Definition : __property TTFDPFields *FieldAssignment = {read=FFieldAssignment, write=SetFieldAssignment}

Type: published

FieldNames

Description: A list of strings specifying the names of the text file fields. If the property FirstLineHeader is true these values are automatically retrieved from the first line of the textfile.

Definition : __property TStringList *FieldNames = {read=FFieldNames}

Type: published

StreamState

Description: Indicates the state of the text stream. Possible values are tssClosed (stream is closed), tssRead (stream is open for read operation), tssWrite (stream is open for write operation). The StreamState could also be set programatically (e.g. to tssRead) to retrieve the textfile fields from the textfile before any action is initiated by the connected TMemTableEC component.

Definition : __property TextStreamState StreamState = {read=FStreamState, write=SetStreamState}

Type: published

Delimiter

Description: Specifies the delimiter used to seperate the field values/columns.

Definition : __property char Delimiter = {read=FDelimiter, write=FDelimiter, default = ','}

Type: published

FirstLineHeader

Description: If set to true the field names are retrieved from the first line of the stream/file. Otherwise they must be specified by setting the FieldNames property.

Definition : __property bool FirstLineHeader = {read=FFirstLineHeader, write=FFirstLineHeader, default = false}

Type: published

QuoteAll

Description: If set to true all field values will be enclosed in quotes on save operations. Otherwise the field values are analysed and only quoted if necessary (containing special charaters).

Definition : __property bool QuoteAll = {read=FQuoteAll, write=FQuoteAll, default = false}

Type: published

6 Events

SetMemTabFieldValue

Description: This event is fired whenever the value from a textfile field was assigned to a field of the TMemTableEC dataset. Here it is possible to change the assigned value. If you have stated virtual fields in the FieldAssignments property, the assignment of their values can be made inside this event handler. The FieldName parameter contains the TMemTableEC field name of the FieldAssignments property.

Handler : void __fastcall (__closure *TTFDPFieldEvent)(TField *memtabfield,
int textfldidx,
const AnsiString &FieldName,
TStringList *rec)

Type: published

SetTextFileFieldValue

Description: This event is fired whenever the value from a TMemTableEC dataset field was assigned to a textfile field. Here it is possible to change the assigned value. If you have stated virtual fields in the FieldAssignments property, the assignment of their values can be made inside this event handler. The FieldName parameter contains the textfield name of the FieldAssignments property.

Handler : void __fastcall (__closure *TTFDPFieldEvent)(TField *memtabfield,
int textfldidx,
const AnsiString &FieldName,
TStringList *rec)

Type: published

OnOpenStreamEvent

Description: This event is fired before any read or write operation is initiated by the data provider. Here you could assign a TStream or a filename for the textfile containing the data. The state parameter specifies if a read or write operation will be performed.

Handler : void __fastcall (__closure *TOnOpenStreamEvent) (TTextFileDataProviderEC *Sender, TextStreamState state)

Type: published

OnCloseStreamEvent

Description: This event is fired after all read or write operations are finished. Here you could free a TStream you have assigned in the OnOpenStreamEvent handler.

Handler : void __fastcall (__closure *TNotifyEvent) (TObject* Sender)

Type: published

OnReadStreamHeader

Description: This event is fired before the fextfile header will be read. If you have any unusual header information in you stream you could retrieve it here, assign the fieldnames to the FieldNames property and set the stream position to where the data section of your stream begins.

Handler : void __fastcall (__closure *TNotifyEvent) (TObject* Sender)

Type: published

OnWriteStreamHeader

Description: This event is fired before the textfile header will be written. If you have any unusual header information in you stream you could write it here and set the stream position to where the data section of your stream begins.

Handler : void __fastcall (__closure *TNotifyEvent) (TObject* Sender)

Type: published