TCrossTab Component

See Also Properities Methods Events

TCrossTab is a component for displaying the multi-dimensions cross fields result for each single database table or database view (if the database is supported). When a application uses TCrossTab, at runtime, user can use the drag and drop features of TCrossTab to switch the sequence of cross row fields and cross column fields. The crossing fields result will be reflected in the new sequence of cross row and column fields.

At design time, the programmer can set the DatabaseName property to desired database, TableName property to desired table, and CTCrossFields property by picking available table fields and set those fields to Fixed Columns, Fixed Rows and Aggregated value fields. TCrossTab provides a property editor to allow the programmer to set the CTCrossFields property easier. When the Active property is set to 'TRUE',the crossing fields result will be shown on a grid.

The 'TCrossTab' component does not support 'Free-Style SQL'. It only deals with one single table or view. For free-style SQL, please use TCrossSQL component.

Quick Start of TCrossTab
TCrossSQL Component

Quick Start of TCrossSQL
TCrossTab Component

Properties

Active
AggrCaption
AggrRowColumn
Align
Canvas
Col
Color
ColWidths
Confirm
Ctl3D
CTCrossFields
CTOptions
Cursor
DatabaseName
DefaultColWidth
DefaultRowHeight
FixedColor
Font
FPDigits
FPFormat
FPPrecision
LeftCol
LoginParams
LoginPrompt
QuoteIdentifier
Row
RowHeights
Serial
TableName
TopRow
Where

Methods

CellRect
GetColCount
GetColField
GetColHeading
GetNumCrossCols
GetNumCrossRows
GetNumValueFields
GetRowCount
GetRowField
GetRowHeading
GetValField
GetValHeading
GetXCell
Refresh
SetCTFields

Events

OnClick
OnDblClickXCell
OnDrawXCell
OnEnter
OnExit
OnLoaded
OnMouseDown
OnMouseMove
OnMouseUp
OnSelectXCell

Quick Start of TCrossTab Component

Do the following steps and check the result.

Step 1: Set the DatabaseName property to the desired database.
Step 2: Set the TableName property to the desired table from the database you have choosen.
Step 3: Set the CTCrossFields property by using property editor dialog, choose the cross column fields, cross row fields and aggregated value fields.
Step 4: Set the Active property to True.

Now, you should see the results. Run the program, and use drag and drop to switch the cross column fields and cross row fields, different crossing results will be shown.

Quick Start of TCrossSQL Component

Before starting to use the TCrossSQL, make sure you are an experienced user of TCrossTab.
If you are going to do crossing table from a table (table1) in a database (database1).
The cross column fields are (colfield1, colfield2) and the cross row fields are (rowfield1,rowfield2) and the value fields are (sum(valuefield),avg(valuefield)).

Step 1: Set the DatabaseName property to database1.

Step 2: Set the SQL property as

```
select colfield1,colfield2,rowfield1,rowfield2,sum(valuefield),avg(valuefield)
from table1
group by colfield1,colfield2,rowfield1,rowfield2
```

Setp 3: Use CSCrossFields property editor to re-arrange the number of cross column fields, cross row fields and aggregated value fields. At the same time, set the headings for fields, cross row fields and aggregated value fields.

Step 4: Set Active property to True.

TCrossSQL Component

See Also Properities Methods Events

TCrossSQL is a component for displaying the multi-dimension cross fields result for data stored in database. TCrossSQL has similar functionality of TCrossTab. The major difference between the two components is TCrossSQL allows the user to issue free-style SQL to access databases which include standard SQL and stored procedures(for some databases). TCrossSQL will pass user's SQL to database directly and will not do any checking, the results returned from the database will be used for forming a crossing result. Only certain reasonable results can form a crossing table, illegal results will cause unexpected output. When user uses TCrossSQL, he/she should have a basic concept of how to issue a legal SQL in odrer to get the result which the user expected. For most of the time, TCrossTab should be able to satisfy user's need and is recommanded to be used. Only advanced user who need to perform more special query which TCrossTab does not support should TCrossSQL be used.
Certain rules have to be followed when the user uses TCrossSQL. The projection list of users SQL will be used for column cross fields, row cross fields and aggregated value fields. The two properties NumColFields and NumRowFields are used for specifying which fields in projection list will be used for column cross fields, row cross fields and aggregated value fields. In the projection list, the first NumColFields items are used for column cross fields, the NumRowFields items are used for row cross fields and the rest of the items are used for aggregated value fields. TCrossSQL will do some checking against NumColFields and NumRowFields properties and give the user an error message if an illegal combination of the two properties is detected.
For additional functionality, please see TCrossTab.

Properties

Active	Ctl3D	FPPrecision	
AggrCaption	CTOptions	LeftCol	
AggrRowColumn	Cursor	LoginParams	
Align	DatabaseName	LoginPrompt	
Canvas	DefaultColWidth	Row	
Col	DefaultRowHeight	RowHeights	
Color	FixedColor	Serial	
ColWidths	Font	SQL	
Confirm	FPDigits	TopRow	
CSCrossFields	FPFormat		

Methods

CellRect
GetColCount
GetColField
GetColHeading
GetNumCrossCols
GetNumCrossRows
GetNumValueFields
GetRowCount
GetRowField
GetRowHeading
GetValField
GetValHeading
GetXCell
Refresh
SetCSFields

See TStringGrid Component.

See TDatabase Component for more details.

See TTable Component for more details.

Active Property
Boolean Field.

When active property is set to true, TCrossTab or TCrossSQL will access database, get the results and put them into the grid. When active property is set to false, the grid will be reset to empty.

AggrCaption Property
String Field.

AggrCaption property string will be used for aggregate row heading and aggregate column heading if AggrRowColumn property was set to true.

Aggregate Property
Set of (avg,count,max,min,sum).

The Aggregate property can be set as **avg, count, max, min and sum**. The five aggregate functions are supported by most databases. The TCrossTab will depend on the target database to do the aggregation of each cell and TCrossTab will do the aggregation of all rows and columns depending if AggrRowColumn property was set or not.
For some other special aggregation functions which some databases support, TCrossTab can not use it. If the user needs to use those special aggregation functions, TCrossSQL component will allow the user to issue free-style SQL. In this case, row and column aggregation will not be supported by the TCrossSQL component.

AggrRowColumn Property
Boolean Field.

The AggrRowColumn property can be set either to **True** or **False.**
When True, TCrossTab will do the aggregation for each row and column. The AggrCaption will be used for the row and column headings.

Confirm Property
String Field.

When you register the TCrossTab/TCrossSQL component, you will get a set of Serial and Confirm number. If these two fields do not have the right information, the TCrossTab will still work fine, the only difference is the user will get a prompt dialog. Just click OK and your application will keep on runing. The reason for this set of serial and confirm numbers is to allow the user to fully test the TCrossTab/TCrossSQL before they purchase the component.

CSCrossFields Property
String List Field.

The CSCrossFields property is a list of strings . The CSCrossFields include three parts, cross column fields, cross row fields and aggregate value fields.All strings are set by SQL property. After user inputs a valid query to SQL property, CSCrossFields are set. The TCrossSQL provides a property editor for CSCrossFields. User can use the property editor to re-arrange which field will be used for cross column field or cross row field or aggregate value field. The fields sequence is not allowed to be changed.The corresponding heading for each field can be changed in the property editor. Cross column fields are used as fixed columns, cross row fields are used as fixed rows and aggregate value fields will be used as cross results.

CTCrossFields Property
String List Field.

The CTCrossFields property is a list of strings. Each string is a valid database table field name. CTCrossFields include three parts, cross column fields, cross row fields and aggregate value fields.
The TCrossTab provides a property editor for CTCrossFields. User can use the property editor to choose the valid field name from database table or view as cross column fields , cross row fields or aggregate value fields.
Cross column fields are used as fixed columns, cross row fields are used as fixed rows and value field is used as common cell data. In the current release, TCrossTab can do multi-dimension crossing results depending on the cross column fields and cross row fields. The aggregate value fields can include more than one entry. Set the aggregation function first and then choose the value field.
The heading for each cross column field, cross row field or aggregate value fields can be set in the property editor also.

CTOptions Property

Set of (goCTAutoColWidth, goCTColSizing, goCTHeadings, goCTHorzLine, goCTRowSizing, goCTRunTimeDragging, goCTVertLine)

Description

goCTAutoColWidth:
If True,All column widths will be adjusted to fit the longest cell.
goCTColSizing:
If True, each column will be allowed to be resized during run time.
goCTHeadings:
If True, the headings for all cross column fields, cross row fields and value field will be visible.
goCTHorzLine:
If True, all horizon lines will be visible.
goCTRowSizing:
If True, each row will allow resizing during run tim.
goCTRunTimeDragging:
If True, run-time dragging will be allowed.
goCTVertLine:
If True, all vertical lines will be visible.

DatabaseName Property
String Field.

Set this property to the desired database.
See TDatabase Component for more details.

FPDigits Property
Integer Field.

Please see FPFormat for more details.

FPFormat Property
Set of (ffGeneral,ffExponent,ffFixed,ffNumber,ffCurency).

All Cells will be displayed as floating value. The format will depend on FPDigits, FPFormat and FPPrecision properties. For more details, please see FloatToStrF.

FPPrecision Property
Integer Field.

Please see FPFormat for more details.

NullCaption Property
String Field.

The NullCaption property is used for displaying null cells. The default value for NullCaption is empty. User can assign a special string to NullCaption, all null cells will
then be shown with NullCaption value.

QutoteIdentifier Property
Boolean Field.

The QuoteIdentifier property can be set to **True** or **False**.
The SQL generated by TCrossTab will quote all identifiers(field names and table names) if the QuoteIdentifier property is set to **True,** otherwise all identifiers will not be quoted.
For some databases which do not support quoted identifier, the QuoteIdentifier property should be set to **False,** otherwise database will return syntax or similar error message.
If QuoteIdentifier is set to **False**, the multi-word field name will not be supported.

RunTimeDragging Property
Boolean Field.

The RunTimeDragging propery can set to **True** or **False.**
If True, TCrossTab/TCrossSQL will allow the user to use drag and drop to switch crosscolumn fields and cross row fields at runtime. Once the cross column fields and cross row fields are changed, the TCrossTab/TCrossSQL will reflect the corresponding results on the grid.

Serial Property
String Field.

When you register the TCrossTab/TCrossSQL component, you will get a set of Serial and Confirm number. If these two fields do not have the correct information, the TCrossTab/TCrossSQL will still work fine. The only difference is the user will get a prompt dialog. Just click OK and your application will keep on runing.
The reason for this set of serial and confirm numbers is to allow the user to fully test the TCrossTab/TCrossSQL before purchasing the components.

SQL Property
String List Field.

SQL property will be used to access the database. Not all queries can produce a legal crossing result, only a legal query can get a legal crossing results. Here is a general legal query for geting a crossing result.

```
select cross_field_one,cross_field_two,....,aggregated_field_one,aggregated_field_two...
from database_table_one,database_table_two,...
where .....
group by cross_field_one,cross_field_two....
```

Where Property
String List Field.

Where property will be used when crosstab query was issued to the database. User can use the where property to filter all unnecessary data. Since the where property will be a part of the query, the syntax of the where property should follow SQL standard. The keyword **where** is not required for where property.

GetColCount Method

Declaration
function GetColCount: Longint;

Description
Return the total number of columns of TCrossTab or TCrossSQL.

GetColField Method

Declaration
function GetColField(Const index : LoingInt): String;

Description
Return the column cross field specified by index .
Index 0 will return first column cross field.

GetColHeading Method

Declaration
function GetColHeading(Const index : LoingInt): String;

Description
Return the cross column's heading specified by index .
Index 0 will return first cross column's heading.

GetNumCrossCols Method

Declaration
function GetNumCrossCols: Longint;

Description
Return the total number of cross columns(column cross fields) of TCrossTab or TCrossSQL.

GetNumCrossRows Method

Declaration
function GetNumCrossRows: Longint;

Description
Return the total numbers of cross rows(row cross fields) of TCrossTab or TCrossSQL.

GetNumValueFields Method

Declaration
function GetNumValueFields: Longint;

Description
Return the total numbers of aggregate value fields of TCrossTab or TCrossSQL.

GetRowCount Method

Declaration
function GetRowCount: Longint;

Description
Return the total number of rows of TCrossTab/TCrossSQL.

GetRowField Method

Declaration
function GetRowField(Const index : Longing): String;

Description
Return the cross row field specified by index .
Index 0 will return first cross row field.

GetRowHeading Method

Declaration
function GetRowHeading(Const index : Longing): String;

Description
Return the cross row's heading specified by index .
Index 0 will return first cross row's heading.

GetValField Method

Declaration
function GetValField(Const index : Longing): String;

Description
Return the value field specified by index .
Index 0 will return first value field.

GetValHeading Method

Declaration
function GetValHeading(Const index : Longing): String;

Description
Return the value heading specified by index .
Index 0 will return first value heading.

GetXCell Method

Declaration
function GetXCell(Const ACol,ARow : Longing): String;

Description
Return the cell string value of TCrossTab/TCrossSQL specified by ACol and ARow
ACol: Column index.
ARow: Row index. .

ReFresh Method

Declaration
procedure ReFresh;

Description
When ReFresh method is called. TCrossTab/TCrossSQL will re-access the database and get a new set of data.

SetCTFields Method

Declaration
function SetCTFields(Const FieldList : TStrings;Const HeadingList : TStrings; NCols, NRows, NVals: integer) :Boolean;

Description
SetCTFields method allows the user to reset the CTCrossFields property. The CTCrossFields property includes three parts, column cross fields, row cross fields and aggregate value fields. At design time, the CTCrossFields property editor allows the user to setup the three parts more easily. Anyhow, when user try to reset the CTCrossFields property at run time, the user should use this SetCTFields method to set the all three parts of the fields.

FieldList: Tstrings.
List of fields strings which include three parts, column cross fields, row cross fields and aggregate value fields.
HeadingList: Tstrings.
List of heading strings which correspond with FieldList.
NCols: Integer.
number of column cross fields in FieldList.
NRows : Integer
number of row cross fields in FieldList.
Nvals : Integer
number of aggregate value fields in FieldList.

SetCSFields Method

Declaration
function SetCSFields(Const HeadingList : TStrings; NCols, NRows, NVals: integer) :Boolean;

Description
SetCSFields method allows the user to re-arrange the fields of the CSCrossFields property and assign new headings for each field. The CSCrossFields property includes three parts, column cross fields, row cross fields and aggregate value fields. All fields are set by the SQL property. SetCSFields can only re-arrange the numbers of each cross fields or value fields. At design time, the CSCrossFields property editor allows the user to re-arrange the three parts more easily. Anyhow, when user try to re-arrange the CSCrossFields property at run time, the user should use this SetCSFields method to re-arrange all three parts of the fields.

HeadingList: Tstrings.
List of strings which should correspond with all three parts, column cross fields, row cross fields and aggregate value fields. The total number of headings should match the total number of fields which are set by the SQL property, if **nil,** no headings will be set. The original value will be used.

NCols: Integer.
number of column cross fields in FieldList.

NRows : Integer
number of row cross fields in FieldList.

Nvals : Integer
number of aggregate value fields in FieldList.

OnSelectXCell Event

Declaration
property OnSelectXCell: TSelectXCellEvent;

Description
The OnSelectXCell event occurs when the user selects a cell in a TCrossTab/TCrossSQL. Use the OnSelectXCell event handler to write the code that handles the selecting of a cell.

TSelectXCellEvent = **procedure** (Sender : Tobject; Col, Row : Longint) **of object**;

OnDblClickXCell Event

Declaration
property OnDblClickXCell: TSelectXCellEvent;

Description
The OnDblClickXCell event occurs when the user double clicks a cell in a TCrossTab/TCrossSQL. Use the OnDblClickXCell event handler to write the code that handles the selecting of a cell. When the user double clicks a cell, the OnSelectXCell event is generated also.

TSelectXCellEvent = **procedure** (Sender : Tobject; Col, Row : Longint) **of object**;

OnDrawXCell Event

Declaration
property OnDrawXCell: TDrawCellEvent;

Description
The OnDrawXCell event occurs after each cell is drawn in a TCrossTab/TCrossSQL. Use the OnDrawXCell event handler to customize cells.

OnLoaded Event

Declaration
property OnLoaded: TNotifyEvent;

Description
The OnLoaded event occurs after the TCrossTab/TCrossSQL is loaded in run time . Use the OnLoaded event handler to write the code that handles the extra things you need to do after the TCrossTab/TCrossSQL is loaded.

See all components.