GridOne

User Guide and Introduction

GridOne 4.0 Open Source under GNU GPL License.

Actsone Company

Http://actsone.co.kr
http://vn.actsone.co.kr/main.cfm

(+84) 83-715-8067

Fax: 02-575-4695

12/27/2013

GridOne 4.0

User Guide

User Guide for Beginner

Version : V4.0 open source under GNU GPL License

Last changed : 27 December 2013

Contents

1. User Manual

This introduction explains about some techniques and information that will be useful to make the best use of GridOne.

Working with GridOne mostly about

- Embed GridOne in HTML page
- Communication with GridOne's methods
- Call GridOne's function
- Set GridOne's style properties
- Set data to dataGrid
- Get data from GridOne
- Use event in GridOne

1.1. How to embed GridOne in HTML page?

GridOne is build from flex project, so the final source is GridOne.swf file which can easy to embed in HTML page. GridOne provide the javascript library for user as bridge connect between GridOne and User's HTML.

To use GridOne user need:

- GridOne.swf
- GridOne.js
- User's HTML file

In GridOne.js code

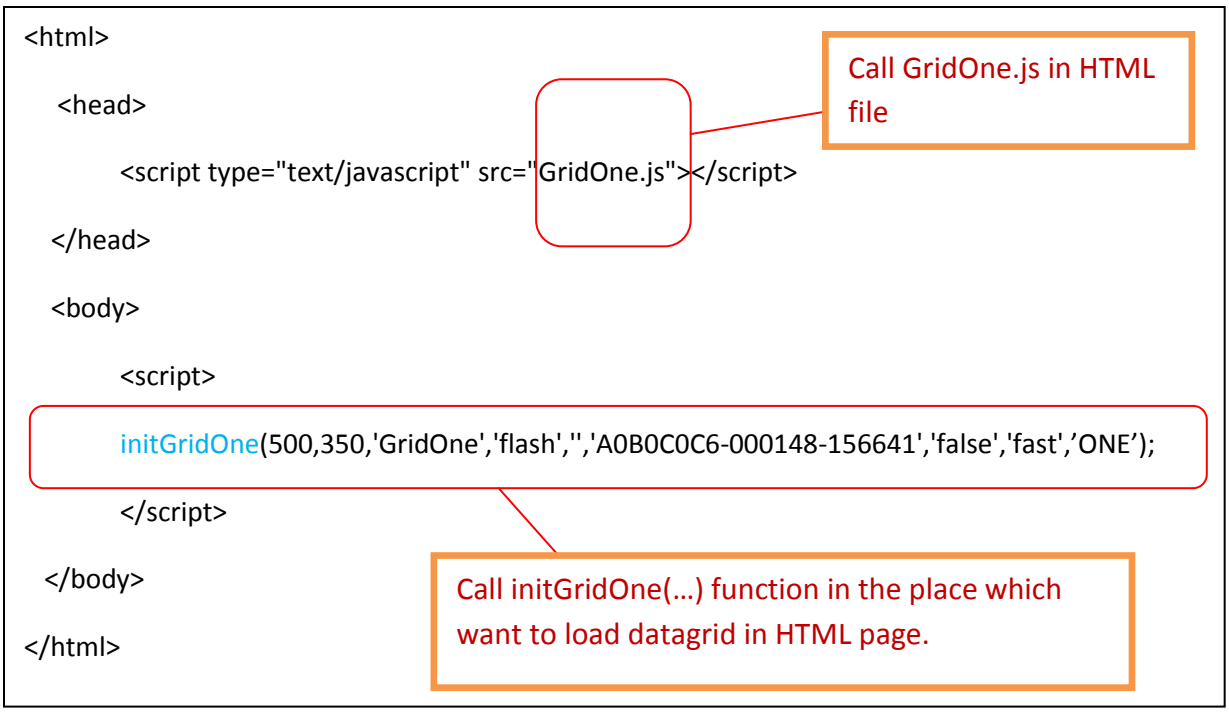
```
Function
initGridOne(width,height,objName,bridgeName,xmlConfig,serialKey,debugMode,performanceMode,containerid)

{

……

}
```

Call initGridOne (…) function in User's HTML file.



initGridOne (…) parameter explain

Parameters	Explain
width	Number of grid's width or percent of grid's width
Height	Number of grid's height or percent of grid's height
objName	String name of grid. EX: " GridOne1"
bridgeName	String of bridge name. In current version "flash" is normal used.
xmlConfig	String of xml configure, If user want reconfigure GridOne. In current version is not normal used.
serialKey	String of serial key specification that provide by Actsone company. "'A0B0C0C6-000148-156641" is true serial key.
debugMode	String of debug mode "true", "false". Normal use is "false" for good performance.
performanceMode	String of performance mode such as "normal", "fast".
containerid	String of grid's container id or name. It important when user use multiple grids in one HTML page. If user set this contain id, user must add this id to initializeHandler function, and boundHeaderComplete event.

	Ex: containerid =”ONE”. ONEinitializeHandler(){} ONEboundHeaderCompleteHandler(event){}

1.2. How to use GridOne.swf?

After embedded GridOne.swf in HTML, all the remain work is about javascript coding block. Now start creates grid’s header, grid’s body, grid’s styles and grid’s data by call GridOne’s functions.

Take a look at javascript the main structure below: User.html

```
<html>
<head>
        <script type="text/javascript" src="GridOne.js"></script>
</head>
<script type="text/javascript">
        var grid;
        function ONEinitializeHandler()
        {
                grid = document.GridOne;
                grid.createEvent("boundHeaderComplete","ONEboundHeaderCompleteHandler");
                setHeader(grid);
        }
        function ONEboundHeaderCompleteHandler()
        {
                setData(grid);
        }
</script>
<body>
        <script type="text/javascript">
                initGridOne(500,350,'GridOne','flash','','A0B0C0C6-000148-156641','false','fast','ONE');
        </script>
```

GridOne object, use for call all functions inside GridOne.swf

GridOne name which provide by user in initGridOne (…) function. If you want use two or more GridOne in one page, you must provide its difference name.

GridOne’s container ID, it must be a string only. Use for distinguish multiple GridOne in one HTML page. Ex: “ONE”, “TWO”,”FIRST”,…. Based on user define.

1.3. How to call functions in GridOne?

Above we explain about the important structure in User.html, that we can see 1 function in block javascript code.

- ONEinitializeHandler() is defined in GridOne.js file. This function is used for start everything such as call setHeader function , create event and set some initial styles.
- boundHeaderCompleteHandler(event) is a event defined in GridOne.swf, which dispatch when header was completed bounding. Normal we use this event to set initial data loading to DataGrid.

Remember use grid object variable when ever call GridOne's functions.
Example : grid.createEvent("boundHeaderComplete","ONEboundHeaderCompleteHandler");

Below is the importance functions list for create DataGrid with GridOne.

Functions	Description
(containerID)**initializeHandler**	is defined in GridOne.js file. It will collection all start functions inside.
addHeader (columnKey, columnHeaderText, columnTyple, maxLenght, columnWidth, visible)	Use to set and create datagrid header. -columnKey is column Key will use to it style -columnHeaderText is column header text/tile -maxLenght is limit to data length for that column -columnWidth is column's width number, or string percent. Visible is editable or not. False will not allow edit data in that column. True is allowed to edit. Ex : grid.addheader("crud","CRUD","crud",20,100,true);
boundHeader()	This function will call when addHeader all complete. It will start create and bound it together as row and column.
createEvent(eventName, eventHandler)	This function will use for call all event inside GridOne.swf. -eventName is event name that is defined in GridOne. -eventHandler is event handler to get data from that event. Ex: grid. createEvent("boundHeaderComplete","boundHeaderCompleteHandler"); See events list go API demo
setProperty(propertyName,propertyValue)	This function will use for set property and style to all columns in datagrid.
setTextData(dataVariable)	This function is use to set data to datagrid as text format. Ex: gridOne.setTextData(textData);

	More detail see API demo
setGridData(dataVariable) **or** **setJSONData**(dataVariable)	This function is used to set json data to datagrid. More detail go API demo
setXMLData(dataUrl)	This function is used to set xml data to datagrid More detail go API demo
setColumnProperty(…)	This function is used for set property or style to a specific column. More detail go API demo

Event	Description
boundHeaderComplete	This event is used to set data for initial load to datagrid.

1.4. How to set Properties for dataGrid?

There are 2 ways to set property for dataGrid:

1) Use setProperty() or setColumnProperty() which explained in the table above.
2) Use function which provided by GridOne for a specific row, column or cell purpose.
   Ex: grid.setRowBgColor(…);
   Grid.setColHDBgColor(…);
   Grid.setColCellBgColor(…);

To know more functions support in GridOne go chapter 3 features list or go here. In GridOne, properties and style are always used separately such as use for header, all columns, one column, all rows, one row, all cells, one cell and for datagrid. User can easy to notify by look at the key word in that function's name.

Key words	Explain
"set"	Express the function is void type
"get"	Express that function is return type
"str"	Express that it is the style with value is String
"n"	Express that it is the style with value is Number.
"b"	Express that it is the style with value is Boolean.
"HD"	Express that property or method is used for header only.
"col"	Express that property or method is used for

	column only.
“row”	Express that property or method is used for row only.
“cell”	Express that property or method will apply to all cells.
“colcell”	Express that property or method is used for on specific cell only.
“combo”	Express that property or method is used for combo only.
“groupMerge”	Express that property or method will apply to column cell merge.
“Grid”	Express that property or method will apply to the hold dataGrid.
“SummaryBar”	Express that property or method will apply to summary bar only.
“Tree”	Express that property or method will apply to tree only.
Other	It will express that property or method based on its meaning.

1.5. How to set data to dataGrid?

Data is the main purpose of GridOne project. GridOne provide the efficiency for user to work with data online, which various data type such text format, json data, xml data, protocol data. And easy to connect to database with many technologies up on user choose.

As mention before, in GridOne provide 4 functions to set data to dataGrid such as

1) setTextData(data)
2) setJSONData(data) or setGridData(data)
3) setXMLData(data)
4) setProtocolData(data) (not popular use)

Now let talk about each data format structure and remember that if the data was set in wrong format structure, load data to dataGrid will be fail. Normal it will not show up any error message but only the empty dataGrid.

Below code will show each data format structure:

1) Text Data Format

Assume that user already set headers for dataGrid which have column Key or (column ID) as below:

- **id** ➔ columnType=”checkbox”
- **dob**➔columnType=”date”
- **phn**➔columnType=”text”
- **name**➔columnType=”text”
- **salary**➔columnType=”number”
- **idnumber**➔columnType=”text”
- **email**➔columnType=”text”

There are some rules for set text format:

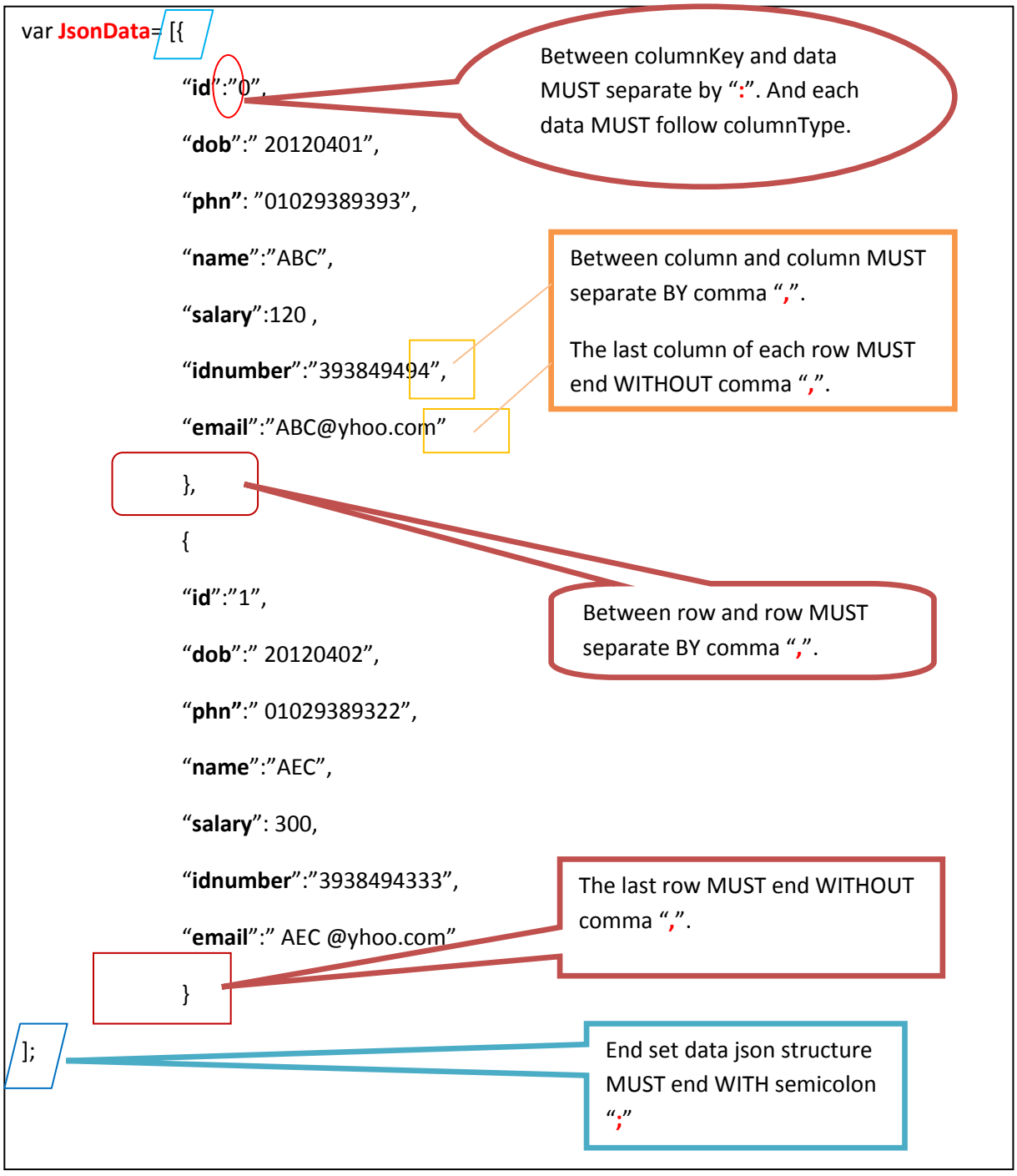
- Each column MUST separate by “|”.
- The first row string is for header column key
- At the end of each row MUST end up with “%%” and “+”
- Each data type MUST follow column type
- The last row MUST end up without “%%”, but with semicolon “;”.

```
var textData = "id|dob|phn|name|salary|idnumber|email %%"
+ "0|20120401|01029389393|AB|120|39382738|lucy@yahoo.com%%"
+ "1|20120401|01029389393|AB|144|39382738|lucy@yahoo.com%%"
+ "0|20120401|01029389393|BC|156|39382738|lucy@yahoo.com%%"
+ "1|20120401|01029389393|AB|276|39382738|lucy@yahoo.com%%"
+ "0|20120401|01029389393|AB|300|39382738|lucy@yahoo.com";
Function setData(grid)
{
        grid.setTextData(textData);
}
```

2) JSON Data Format

JSON data is most efficiency for user to work with because its structure clean and look similar with array object. In case good performance and much stable, we strongly recommend user to use JSON data format with GridOne.

Assume that user set header same as in 1). You will make JSON like this:



```
Function setData(grid)
{
        grid.setJSONData(JsonData);
}
```

3) XML Data Format

XML data format is good loading performance but we don't recommend user use it, if not necessary because another side it make big problem with vertical scrolling performance.

XML data format is really simple and easy to understand. Please look at a sample below: User.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<data>
<item>
        < id >0</ id >
        < dob >20120402</ dob >
        < phn >01029389322</ phn >
        < name > AEC </ name >
        < salary >300</ salary >
        < idnumber >3938494333</ idnumber >
        < email > AEC @yhoo.com </ email >
</item>
<item> ……..</item>
</data>
```

This is the main root of all items.

One item = one row data object in dataData.

This element is stand for column key.

XML data file always set at server site. GridOne set XML data by use HTTP Service provider to load data from server.

```
Function setData(grid)
{
        grid.setXMLData("http://..../User.xml");
}
```

1.6. How to get data from dataGrid?

GridOne provide many functions to get data from dataGrid. And each function is used for specific purpose. Among of those functions, we collected many functions that often use below.

Functions	Description
getGridData (more detail go here)	Use to get JSON data from dataGrid.
getXMLData (more detail go here)	Use to get XML data from dataGrid.
getTextData (more detail go here)	Use to get Text data from dataGrid.
getArrayData	Use to get data as array from dataGrid.
getItemAt	Use to get data from one specific row.
getCellValue (more detail go here)	Use to get data from one specific cell.
getDataGridString (more detail go here)	Use to get Protocol data from dataGrid.

1.7. How to use event in GridOne?

GridOne supported many events included internal event and custom event. When work with event, user can get start with "createEvent(…)" function. For efficiency for user, GridOne provide only this function to register event which will be listen.

Normally except "boundHeaderComplete" event, user can call it any place in HTML block code. Each event has its own return values. More detail visit GridOne API.

Below is example work with event: let skip "boundHeaderComplete" event.

```
function initializeHandler()
{
        gridOne = getGrid();
        gridOne.createEvent("boundHeaderComplete","boundHeaderCompleteHandler");
        gridOne.createEvent("onCellClick","onCellClickHandler");
        setHeader(gridOne);
}
Function onCellClickHandler (event)
{
        Alert(event.nRow);
}
```

This event name is provided by GridOne. User can't change this name.

This handler will return the event value every time when its dispatch.

nRow is return parameter from that event.

1.8. GridOne Interface Interaction with User

1.8.1. Header drag and drop

Click on one header and move it to another column and then do release mouse.

1.8.2. Header Sort Action

No.	City/Province	Nation	Continent
10	Abidjan	Ivory Coast	Africa
2	Da Nang	Viet Nam	Asia

Click on one header, the sort icon will show up, and the sort function will perform again and again.

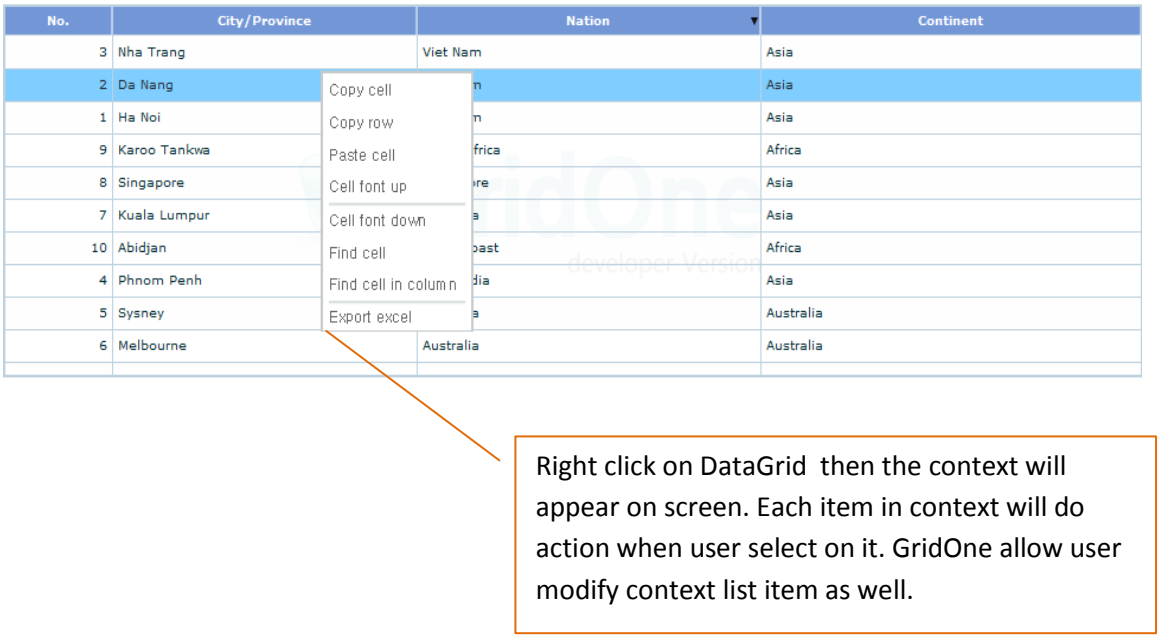
1.8.3. Header Resize Column Action

Put mouse over the border of header and when the icon show up, click and drag to left or right direction. The column will automatic change width size.

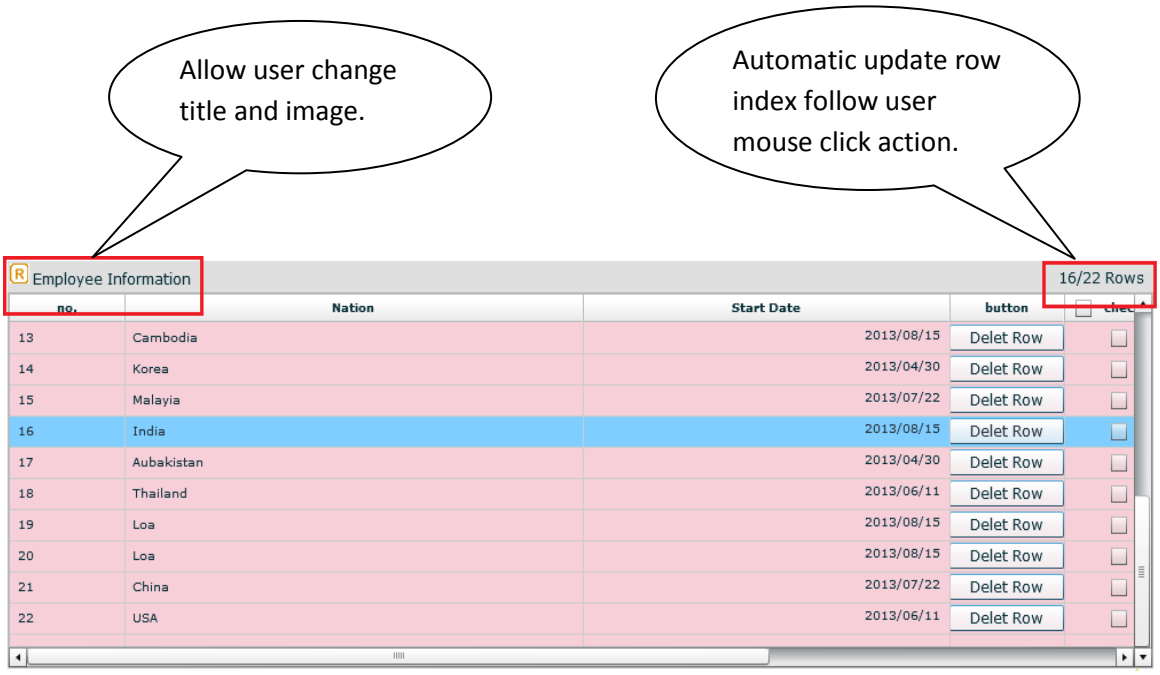
No.	City/Province	Nation	Continent
10	Abidjan	Ivory Coast	Africa
2	Da Nang	Viet Nam	Asia
1	Ha Noi	Viet Nam	Asia
9	Karoo Tankwa	South Africa	Africa
7	Kuala Lumpur	Malaysia	Asia
6	Melbourne	Australia	Australia
3	Nha Trang	Viet Nam	Asia
4	Phnom Penh	Cambodia	Asia
8	Singapore	Singapore	Asia

No.	City/Province	Nation	Continent
10	Abidjan	Ivory Coast	Africa
2	Da Nang	Viet Nam	Asia
1	Ha Noi	Viet Nam	Asia
9	Karoo Tankwa	South Africa	Africa
7	Kuala Lumpur	Malaysia	Asia
6	Melbourne	Australia	Australia
3	Nha Trang	Viet Nam	Asia
4	Phnom Penh	Cambodia	Asia
8	Singapore	Singapore	Asia
5	Sysney	Australia	Australia

1.8.4. Context Menu Support



1.8.5. GridOne Header Support



Next chapter, we will present GridOne general concept and provide some real example.

2. GridOne General Concept and Example

2.1. GridOne Introduction

2.1.1. What is GridOne?

GridOne is a grid solution which develops by Actsone Company. GridOne is a complete framework to show content in a very natural way dataGrid.This also is called flex application technology and making use of Adobe Flash technology.

Published GridOne, i.e. GridOne made available on a server, can be viewed by everybody with internet access and the Flash Plugin installed. Most users (>99%) can view dataGrid immediately. Those without installed plugin can download the plugin for free; a link is provided in case GridOne detects Flash is not installed.

The basic concept of GridOne is provides a good solution for user working online with data which feel similar work with Microsoft Excel. If user already familiar with Microsof Excel, this will be every fast to understand GridOne process flow.

2.1.2. What is benefit of GridOne?

GridOne provide many benefit to user such as:

- ✓ Online data management
- ✓ Connect database with working data.
- ✓ Easy to embed in user html page.
- ✓ Easy to print as PDF
- ✓ Easy to export data to Excel or CSV
- ✓ Easy to import data from Excel or CSV
- ✓ Easy to sort data, filter data, search data
- ✓ Easy to insert, delete and update data.
- ✓ Easy to use summary bar with data
- ✓ Support many data format such as JSON, TEXT, XML, PROTOCOL data.
- ✓ Easy to use tree data
- ✓ Easy to merge row data

- ✓ Easy to set styles
- ✓ Interface good looking than Excel
- ✓ Easy add rows and columns
- ✓ Provide many events for user
- ✓ Fast Scrolling
- ✓ Fast and very flexible with user interaction
- ✓ Support many styles and logic functions
- ✓ Easy to get data from many ways and many return format.
- ✓ Support many columns types
- ✓ Support accessibility plug-in
- ✓ Support many features

2.1.3. Target of GridOne

GridOne is available to many kinds of Companies and Organizations which have IT department.

Target	Usage Purpose
Normal Company	Employee Management Online
Export/Import Product Company	Stock Management Online Customer review report online
IT Company	Use GridOne embed with their product to provide data viewer feature for user.
University /Library	Students Management Online Books List Management Online

2.1.4. GridOne Version List

To improve GridOne and to support more feature to customer, Actsone Company always upgrade and develop continuously. Below is the table list version of GridOne and some description.

GridOne Version	Description
GridOne v2.0	Based on blueOne component and flex dataGrid
GridOne v3.0	Based on flex AdvancedDataGrid, support more features.
GridOne v3.5	Based on flex AdvancedDataGrid, add more features and functions.
GridOne v4.0 (open source)	Based on flex AdvancedDataGrid, add more features and improve scroll performance.
GridOne v5.0 (open source)	Based on flex AdvancedDataGrid, change structure

	design to enhance fast loading with big data, and fast scrolling performance.

Anyway, GridOne is a solution product, not stand-alone product. It needs developer to understand and have experience work with javascript and HTML technology.

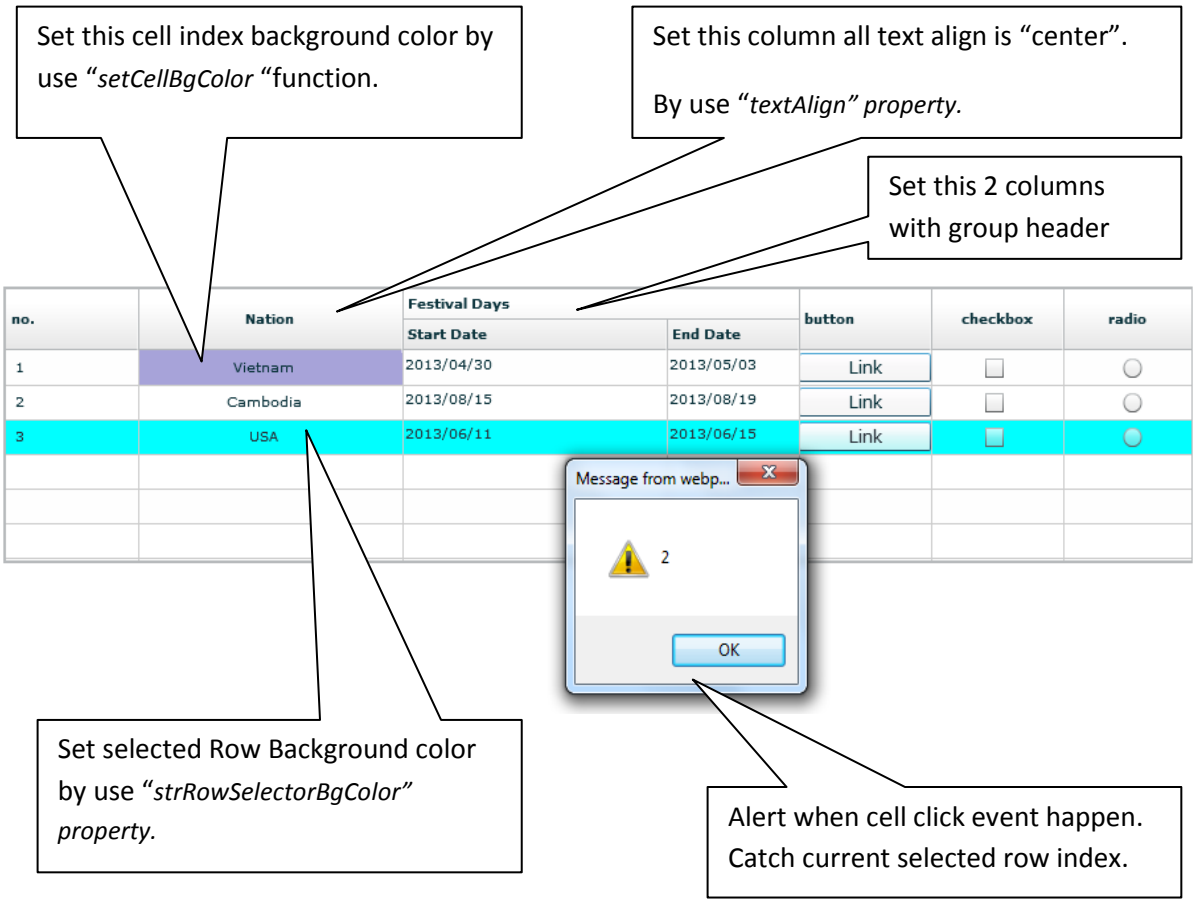
In the next section, we turn back to GridOne usage one again for help to understand more about GridOne.

2.2. HTML, Javascript and Flex, All together

In chapter 1 user manual, we almost explain about how to get start with GridOne. In here we continue, but more deeply to GridOne's API.

As explain above, In user.html file have divide into 2 parts javascript part in head tag will contain GridOne initializeHandler, and other functions and body part will put GridOne screen.

Below is the hold example code: User.html



```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>Demo GridOne</title>

<script type="text/javascript" src="GridOne.js"></script>

<script type="text/javascript">

var gridOne;

/*get gridone*/

function getGrid() {

   return document.GridOne;                    //it may be customized again

}

/*********************************************init grid***************/

/*init for grid one. This function is called by GridOne internally */

function ONE initializeHandler()

{

            gridOne = getGrid();

            gridOne.createEvent("boundHeaderComplete"," ONEboundHeaderCompleteHandler");

            gridOne.createEvent("onCellClick","onCellClickHandler");

            setHeader(gridOne);

            setProperties(gridOne);

}

/*This function is called automatically after headers and columns are created.*/

function ONEboundHeaderCompleteHandler()

{

             setData(gridOne);

`
```

Get GridOne object id

Catch event from GridOne

Call to set data function when header bound complete.

```
function onCellClickHandler(e)
{
        alert(e.nRow);
}
//set header in dataGrid
//=====================================================
 function setHeader(gridOne)
 {
        gridOne.addHeader("no","no.","autonumber",20,100,false);
        gridOne.addHeader("nation", "Nation", "text", 20,200, false);
        gridOne.addHeader("startDate", "Start Date", "date", 20, 200, false);
        gridOne.addHeader("endDate", "End Date", "date", 20,100, false);
        gridOne.addHeader("button","button","button",20,100,true);
        gridOne.addHeader("check","checkbox","checkbox",20,100,true);
        gridOne.addHeader("radio","radio","radiobutton",20,100,true);

        gridOne.addGroup("festivalData", "Festival Days");
        gridOne.appendHeader("festivalData","startDate");
        gridOne.appendHeader("festivalData","endDate");
        gridOne.boundHeader();
 }
//set  data as json format
var flatJSONData = [{"nation": "Vietnam","startDate": "20130430","endDate": "20130503"},
            {"nation": "Cambodia","startDate": "20130815","endDate": "20130819"},
            {"nation": "Korea","startDate": "20130430","endDate": "20130503"}];
```

Catch event return data.

nRow=Current cell row index

Start set header to GridOne

Set dataGrid header group

More detail go here.

```
//set data
 //=========================================================
 function setData(gridOne)
 {
 //set data into dataGrid
  gridOne.setGridData(flatJSONData);
 }
//set properties and styles
function setProperties(grid)
{
grid.setProperty("strRowSelectorBgColor","0x00FFFF");
grid.setColumeProperty("nation","textAlign","center");
grid.setCellBgColor ("nation",0,"0xA7A3D9");
}
</script>
</head>
<body>
        <div>
                <script type="text/javascript">
                 initGridOne(800,350,'GridOne','flash','','A0B0C0C6-000148-156641','false','fast','ONE');
                </script>
         </div>
</body>
</html>
```

Set JSON data to dataGrid.

Here set properties with 3 purposes:

-set for general use case

-set for specific column

-set for specific cell

Display GridOne's screen in here.

3. Features List GridOne 4.0

3.1. Special Features List

In this section, we will collection special features which current support customer and include some description for each feature.

Special Features	Description
Row and column fix (FreezePanes)	Allow user to set row fix or column fix which scrolling event. See here
Draw cell border	Allow user to draw cell border with color and thickness border. See here
Accessibility support	Allow user use accessibility tool. See here
ToolTip support	Allow user use tooltip with -column tooltip. See here -vertical scroll tooltip. See here -set column information tooltip. See here
printPDF	Allow user to print DataGrid to pdf file. See here
Excel Export	Allow user to export DataGrid to excel/csv file. See here
Excel Import	Allow user to import DataGrid to import excel/csv file to DataGrid. See here
Context Menu	Allow user use context menu and modify context item and function. See here
Search data	Allow user to do search data in DataGrid. See here
Filter data	Allow user to do filter data in DataGrid. See here.
Mask input	Allow user set mask input to data. See here
Summary Bar	Allow user set total and subtotal column and row. See here

Tree	Allow user data as tree format. See here
Thousands separator	Allow user set thousands separator to data. See here.
CRUD	Allow user get status of data when data was added, delete, or update. See here
Progress bar	Allow user to set progress bar. See here
Multiple language	Allow user to change language. Current version we support 3 languages (Korean, English, Chinese). See here
Multiple dataGrids in one page	Allow user create and use many dataGrid in one html page at same time. See here

3.2. Column Type List

In this version, GridOne support 27 column types. Below is the table list all column types and some description.

Column Types	Description
IMAGETEXT	Column available for data text with image.
IMAGETEXTRIGHT	Column available for with data text with image at right.
IMAGE	Column available for only image.
HIDEIMAGETEXTRIGHT	Column available for data text with image hidden at right.
CHECKBOX	Column available for data checkbox renderer.
DATE	Column available for data text as date and calendar.
TEXT	Column available for data text.
AUTONUMBER	Column available for data as auto number.
COMBO	Column available for data text with combo renderer.

COMBODYNAMIC	Column available for data with hidden combo renderer.
MULTICOMBO	Column available for data as multiple combo connection.
MULTICOMBOBOX	Column available for data with multiple combo box item renderer.
NUMBER	Column available for data as number only.
RADIOBUTTON	Column available for data as radio button renderer.
CRUD	Column available for data text as CRUD only. This column will notice all activity from other column and generate data itself.
TREE	Column available for data with Tree only.
BUTTON	Column available for data as button renderer.
TIME	Column available for data text as time only.
DATETIME	Column available for data as date and time.
LINK	Column available for data as link connection only.
PERCENT	Column available for data as percentage only.
COMBOHEADER	Column available for header have combo renderer.
HTML	Column available for data as HTML.
HTMLHEADER	Column available for header set data as HTML.
TOTAL	Column available for data as total.
TEXTAREA	Column available for data with text area renderer.

3.3. Methods Support List

In this section, we will list all methods which current support customer and include some description for each methods.

Methods	Description
addHeaders	Add more than one header to DataGrid in one time. More detail go here.
insertHeader	Insert one header to DataGrid. More detail go here
setColHDAlign	Set text-align for column header. More detail go here
getColHDKey	Get column header key (ID). More detail go here
getColHDText	Get column header title. More detail go here
setColHDText	Set column header title. More detail go here
getColHDIndex	Get column header index. More detail go here
setColHDBgColor	Set background column to column header. More detail go here
setColHDFgColor	Set font color to column header title. More detail go here
getColHDVisibleKey	Get column key by column index. More detail go here
getColHDVisibleIndex	Get column index by column key. More detail go here
setColHDCheckBoxVisible	Set checkbox column header visible checkbox renderer or not. More detail go here
setGroupHDText	Set title to group header. More detail go here
getGroupHDText	Get title from group header. More detail go here
setGroupHDColor	Set background color to group header. Set More detail go here
setGroupHDFont	Set font name to group header title. More detail go here

createGroup	Create new group header. More detail go here
setHeaderRightPadding	Set padding to header title from right. More detail go here
setDateFormat	Set date format for data in date column. More detail go here
setNumberFormat	Set number format for data in number column. More detail go here
setCRUDMode	Set CRUD mode for data in crud column. More detail go here
clearCRUDMode	Clear all set CRUD mode. More detail go here
cancelCRUDRow	Cancel CRUD from specific row. More detail go here
cancelCRUD	Cancel CRUD from all rows. More detail go here
getColumns	More detail go here
setColHide	More detail go here
getColWidth	More detail go here
setColCellActivation	More detail go here
getColCellActivation	More detail go here
setColFix	More detail go here
resetColFix	More detail go here
setColHDCheckBoxValue	More detail go here
setColWidth	More detail go here
getColCount	More detail go here
setColCellSort	More detail go here
setColCellAlign	More detail go here
setColCellMerge	More detail go here
setImagetextAlign	More detail go here
getActiveColKey	More detail go here
getColMaxLength	More detail go here
getColType	More detail go here
setColCellBgColor	More detail go here
setColCellFgColor	More detail go here
setColCellFont	More detail go here
setColCellFontBold	More detail go here
setColCellFontCLine	More detail go here

setColCellFontItalic	More detail go here
setColCellFontName	More detail go here
setColCellFontSize	More detail go here
setColCellFontULine	More detail go here
setColCellRadio	More detail go here
changeColumnSeparator	More detail go here
setColIndex	More detail go here
isColHide	More detail go here
setColumnProperty	More detail go here
setCellPaddingRight	More detail go here
setCellPaddingLeft	More detail go here
setCellBorderColor	More detail go here
removeCellBorderAll	More detail go here
removeCellBorder	More detail go here
addRow	More detail go here
insertRow	More detail go here
deleteRow	More detail go here
moveRow	More detail go here
setRowHide	More detail go here
isRowHide	More detail go here
getRowCount	More detail go here
setActiveRowIndex	More detail go here
getActiveRowIndex	More detail go here
setRowActivation	More detail go here
setRowBgColor	More detail go here
setRowFgColor	More detail go here
search	More detail go here
filter	More detail go here
getCellValueIndex	More detail go here
setCellValueIndex	More detail go here
getCellHiddenValueIndex	More detail go here
setCellHiddenValueIndex	More detail go here
getCellHiddenValue	More detail go here
setCellHiddenValue	More detail go here
setCellValue	More detail go here
getCellValue	More detail go here
setCellImage	More detail go here
getCellImage	More detail go here
setCellBgColor	More detail go here
setCellFgColor	More detail go here
setCellFont	More detail go here
setCellFontBold	More detail go here
setCellFontCLine	More detail go here

setCellFontItalic	More detail go here
setCellFontName	More detail go here
setCellFontSize	More detail go here
setCellFontULine	More detail go here
setCellFocus	More detail go here
setCellActivation	More detail go here
getCellActivation	More detail go here
setComboJSONData	More detail go here
addComboList	More detail go here
addComboListValue	More detail go here
getComboListKey	More detail go here
getComboSelectedListKey	More detail go here
getComboListCount	More detail go here
getComboHiddenValue	More detail go here
getComboText	More detail go here
getComboSelectedIndex	More detail go here
setComboSelectedIndex	More detail go here
setComboSelectedHidden	More detail go here
getComboSelectedHidden	More detail go here
setComboRowCount	More detail go here
hasComboList	More detail go here
clearComboList	More detail go here
addImageList	More detail go here
setCoCellImage	More detail go here
removeImageList	More detail go here
getImageListURL	More detail go here
setImageListSize	More detail go here
clearImageList	More detail go here
getImageListCount	More detail go here
addGridImageList	More detail go here
setColCellGridImageList	More detail go here
clearGridImageList	More detail go here
setGridImageListSize	More detail go here
setGroupMerge	More detail go here
isGroupMergeColumn	More detail go here
hasGroupMerge	More detail go here
clearGroupMerge	More detail go here
addSummaryBar	More detail go here
clearSummaryBar	More detail go here
getSummaryBarValue	More detail go here
hasSummaryBar	More detail go here
setSummaryBarColor	More detail go here
setSummaryBarFont	More detail go here

setSummaryBarFormat	More detail go here
setSummaryBarFunction	More detail go here
setSummaryBarText	More detail go here
setSummaryBarValue	More detail go here
clearExcelInfo	More detail go here
excelExport	More detail go here
excelImport	More detail go here
setColCellExcelAsterisk	More detail go here
setExcelHeader	More detail go here
setExcelFooter	More detail go here
excelExportByActiveX	More detail go here
excelImportByActiveX	More detail go here
setTreeMode	More detail go here
collapsTreeAll	More detail go here
collapseTreeNode	More detail go here
deleteTreeNode	More detail go here
expandTreeAll	More detail go here
expandTreeNode	More detail go here
getRowIndexFromTreeKey	More detail go here
getTreeChildNodeCount	More detail go here
getTreeChildNodeKey	More detail go here
getTreeFirstNodeKey	More detail go here
getTreeKeyFromRowIndex	More detail go here
getTreeNextNodeKey	More detail go here
getTreeNodeDepth	More detail go here
getTreeParentNodeKey	More detail go here
getTreePrevNodeKey	More detail go here
getTreeSummaryValue	More detail go here
hasTreeChildNode	More detail go here
hasTreeNextNode	More detail go here
hasTreeParentNode	More detail go here
hasTreePrevNode	More detail go here
insertTreeNode	More detail go here
isTreeNodeCollapse	More detail go here
isTreeNodeExpand	More detail go here
isTreeNodeKey	More detail go here
moveTreeNode	More detail go here
setTreeClickAction	More detail go here
getCheckedRowsIndex	More detail go here
getCheckBoxValue	More detail go here
clearGrid	More detail go here
strLoadingbarUrl	More detail go here
showBusyBar	More detail go here

closeBusyBar	More detail go here
addDefaultContextMenuItem	More detail go here
addUserContextMenuItem	More detail go here
removeAllContextMenuItem	More detail go here
addContextMenuSeparator	More detail go here
setTextData	More detail go here
setJSONData	More detail go here
setJsonDataUrl	More detail go here
setXMLData	More detail go here
setAccessReader	More detail go here
setAccessReaderHeader	More detail go here
fixedLastColumn	More detail go here
getClientDataString	More detail go here
undoRowHide	More detail go here
setMultiRowsHide	More detail go here
printPDF	More detail go here
setToolTipInfo	More detail go here
setLanguage	More detail go here

3.4. Styles Support List

In this section, we will list all styles which current support and include some description for each style.

Styles / Properties	Description
strRowSelectorBgColor	More detail go here
strAlternateRowsBgColor	More detail go here
strAlternateRowsFgColor	More detail go here
strSelectedCellBgColor	More detail go here
strSelectedCellFgColor	More detail go here
strHDBgColor	More detail go here
strHDFgColor	More detail go here
strCellBgColor	More detail go here
strCellFgColor	More detail go here
strGridBorderColor	More detail go here
strGridBgColor	More detail go here
strActiveRowFgColor	More detail go here
strCellFontName	More detail go here
nCellFontSize	More detail go here
bCellFontBold	More detail go here
bCellFontItalic	More detail go here

bCellFontULine	More detail go here
nHDFontSize	More detail go here
bHDFontBold	More detail go here
bHDFontItalic	More detail go here
bHDFontULine	More detail go here
bCellFontCLine	More detail go here
bHDFontCLine	More detail go here
strHDFontName	More detail go here
strBgImage	More detail go here
strBgImageStyle	More detail go here
bHDVisible	More detail go here
inputMask	More detail go here
useThousandsSeparator	More detail go here
nHDLineSize	More detail go here
nRowHeight	More detail go here
nCellPadding	More detail go here
strHDAlign	More detail go here
strHDSizing	More detail go here
bHDMoving	More detail go here
nHDLines	More detail go here
strHDBorderStyle	More detail go here
bNullValueNumberFormat	More detail go here
strGridBorderStyle	More detail go here
strRowScrollDragAction	More detail go here
strMouseWheelAction	More detail go here
strCellClickAction	More detail go here
strHDClickAction	More detail go here
bHDDblClickAction	More detail go here
buttonText	More detail go here
bContextMenuVisible	More detail go here
bUserDefaultContextMenu	More detail go here
bUserContextMenu	More detail go here
bExternalScroll	More detail go here
allowResizeLastColumn	More detail go here
bManualEditCalendar	More detail go here
alwayKeepFormatCalendar	More detail go here
bUpdateNullToZero	More detail go here
bDisplayZeroToNull	More detail go here
bFilterDataIndex	More detail go here
bAutoWidthColumn	More detail go here
nRowHideBuffer	More detail go here
bAllowResizeDgHeight	More detail go here
nMinContentHeight	More detail go here

checkboxTrueValue	More detail go here
checkboxFalseValue	More detail go here
bResizeHeightByApp	More detail go here

3.5. Events Support List

In this section, we will list all events which current support and include some description for each event.

Events	Description
onCellClick	Get event when user click on cell. This event will return parameters such as : - ***nRow*** : row index click - ***columnKey***: column key click - ***nRowBk***: original row index - ***editable***: status of that column (true/false). More detail go here
onCellChange	Get event when user change data in cell. This event will return parameters such as: - ***nRow***: row index - ***columnKey***: column key - ***strOldValue***: old data in cell - ***strNewValue***: new data in cell More detail go here
onComboChange	Get event when user change combo item. This event will return parameters such as: - ***nRow***: row index - ***columnKey***: column key - ***strOldValue***: old combo item data - ***strNewValue***: new combo item data - ***oldIndex***: old item index - ***newIndex***: new item index - More detail go here
onHDCheckBoxClick	Get event when user click on checkbox header renderer. This event will return parameters such as: - ***columnKey***: column key

	- ***strNewValue***: value of header checkbox (true/false) More detail go here
onHeaderClick	Get event when user click on header. This event will return parameters such as: - ***columnKey***: column key More detail go here
onCellDblClick	Get event when user double click on cell. This event will return parameters such as. - ***nRow***: row index - ***columnKey***: column key More detail go here
onCollapse	Get event when user collapse the tree data. This event will return parameters such as: - ***strTreeKey***: tree key More detail go here
onExpand	Get event when user expend tree data. This event will return parameters such as: - ***strTreeKey***: tree key More detail go here
onCellRClick	Get event when user right click on cell. This event will return parameters such as: - ***nRow***: row index - ***columnKey***: column key More detail go here
onColScroll	Get event when user scroll horizontal bar. This event will return parameters such as : - ***nWidth***: dataGrid current width - ***nLeft***: This value is scroll's bar end of left. - ***nRang***: horizontal scroll track's width More detail go here

onRowScroll	Get event when user scroll vertical bar. This event will return parameters such as : - ***nFirstVisibleRowIndex***: first row index - ***nLastVisibleRowIndex***:last row index More detail go here
onRowActivate	Get event when user select on row. This event will return parameters such as: - ***nRow***: row index More detail go here
onMouseOut	Get event when user mouse out from cell. - ***nRow***:row index - ***columnKey***: column key More detail go here
onMouseOver	Get event when user mouse over on cell. This event will return parameters such as: - ***nRow***: row index - ***columnKey***: column key More detail go here
onEndFileExport	Get event when user download excel file success. More detail go here
beforeShowUserContextMenu	Get event before show user context menu. This event will return parameters such as: - ***nRow***: row index - ***strMenuKey***: menu key - ***columnKey***: column key More detail go here
userContextMenuClick	Get event when user click on context menu. This event will return parameters such as: - ***nRow***: row index - ***columnKey***: column key - ***strMenuKey***: menu key - ***strMenuItemKey***: menu item key More detail go here

onRadiobuttonClick	Get event when user click on radio button. This event will return parameters such as: - ***nRow***: row index - ***strNewValue***: new check box value - ***columnKey***: column key More detail go here
onCheckboxClick	Get event when user click on checkbox item. This event will return parameters such as: - ***nRow***: row index - ***columnKey***: column key - ***strNewValue***: new checkbox value - ***columnIndex***: column index More detail go here
textInputKeyUp	Get event when user use key up on item. This event will return parameters such as: - ***strText***: text edit - ***keyCode***: key code - ***columnKey***: column key - ***nRow***: row index More detail go here
onCellMode	Get event when user start edit on cell. This event will return parameters such as: - ***strMode***: cell activation mode - ***nRow***: row index - ***columnKey***: column key More detail go here
imageIconClick	Use for column type is related to image. Return 2 parameters : - ***columnKey***: column key - ***nRow***: row index
onTreeNodeClick	Use for column type is tree only. Return 2 parameters: - ***strTreeKey***: tree key as string - ***nRow***: row index.
onDgFocus	Dispatch event when mouse focus on dataGrid. Return 1 parameter:

	- ***bridgeName***: bridge name
onClick	Get event when user click on GridOne. This event will return parameter : - ***bridgName***: bridge name
onRowKeyMove	Dispatch event when user use arrow key to move row. Return 1 parameter: - ***newIndex***: new row index
onRowVisible	Dispatch event when data loading complete. Return 1 parameter: - ***totalRow***: total of rows number

4. Online Demo GridOne

Online Demo GridOne API

Online Demo GridOne Showcase

Video about "How to create dataGrid with GridOne" follow links of tab "Guide by Video"

5. GridOne Software (Binaries and Soucre) Download

GridOne 4.0 Software (Trail Package) follows link of tab "1.Download trial".

GridOne4.0 Open-Source (Source Code Package) follows link of tab "GridOne4.0-open-source"

6. Demo package download

GridOne demo download package follows link of tab click on button "DownLoad Demo". Each case has this link.