

InaGrid Frequently Asked Questions (FAQ)

How do I create an InaGrid control in an application?

In Visual Basic 5.0 and above:

To add the InaGrid control to your project, select Project | Components... Select the InaGrid ActiveX Control Module, InaEdit ActiveX Control Module, InaCombo ActiveX Control Module and InaCheck ActiveX Control Module.

The InaGrid controls will now be available in the Visual Basic Toolbox. Select the InaGrid control from the Toolbox and insert it into a form as usual.

In Visual C++ 5.0 and above:

To add the InaGrid control to your project, select Project | Add To Project | Components and Controls... In Registered ActiveX Controls folder select InaGrid Control. The InaGrid control wrapper class will be added to your project.

To insert the InaGrid control in an application window derived from CWnd, implement a WM CREATE message handler in your application window. Inside this function, create the InaGrid control as a child of your window by calling the Create method of the InaGrid control wrapper class.

To insert the InaGrid control in a dialog, choose InaGrid control from the Control toolbar and select the destination in the dialog where the control is to be placed. You can also click the right mouse button in the dialog. Choose Insert ActiveX control... from the popup-menu and then select InaGrid Control. The InaGrid control is inserted into your dialog.

How do I display row and column information?

The InaGrid control presents a virtual view on data in your program. Whereas most list and grid controls maintain a copy of the row and column information within their own internal data structures, the InaGrid control relies on the application to maintain its data – where it exists to begin with. Also, since the InaGrid control does not bind to a data source, it presents the programmer many more options to manipulate and check data before and after it is presented and edited in the control.

Because the InaGrid control presents a virtual view of your program data, it only requires that data be supplied by the application as it displays it onscreen. As the user scrolls through the data presented in the grid, the control asks the application for row and column information that will be needed for each page of data. Your application needs to be ready to respond to these requests as they occur.

The event that calls back into your application is *GetData*. See the sample programs for examples of responding to this event and providing your application's data.

Using 64-bit data types

The large capacity of the InaGrid control is driven by the use of 64-bit numbers in all functions that require a row number. Visual Basic and Visual C++ handle 64-bit quantities differently. The InaGrid control provides a custom data type, *ROWNUMBER*, that is equivalent to a *CURRENCY* data type in Visual Basic and a *CY* data type in Visual C++.

Visual Basic

The **ROWNUMBER** data type must be scaled properly in order to interpret the value. When using a value from a function that returns a **ROWNUMBER**, you must multiply by 10,000. Conversely, to provide a scaled value to a function that expects a **ROWNUMBER** data type, you must divide by 10,000. When passing values to and from functions that expect a **ROWNUMBER** type, no scaling is necessary.

Visual C++

The **ROWNUMBER** data type in Visual C++ is equivalent to an **CY** data type. This requires all accesses to the variable to be through the **int64** union member.

How do I edit data in InaGrid cells?

By default, the InaGrid control is built to provide editing of an InaGrid cell when the user chooses to edit the cell data. The user initiates an edit action by double-clicking on the cell with the mouse, or pressing the F2 key. If an InaGrid cell has the focus, pressing ENTER sets the first editable cell on the row into edit mode. Subsequent presses of the TAB key moves through all of the editable cells in the row in turn, placing each one into edit mode.

When edit mode is entered, the InaGrid control informs the application using the **OnEditCell** event. Setting the classid to 0 informs the InaGrid control to cancel. It is in this event that an OCX can be embedded in an InaGrid cell. After this the InaGrid control fires the **GetEditLic** event allowing for the embedded OCX license key if needed.

The InaGrid control informs the user that the cell edit control has been created using the **OnInitEditCell** event. This allows for modification of the edit cell object before it is shown, or in the case of the InaEdit control, the style can also be changed.

If the user changes data by editing the cell, the InaGrid control informs the application of the changes using the **SetData** event. This provides your application with the new cell information for updating the application's information.

How to use InaGrid?

Create an InaGrid control.
Add Column Headers/PreHeaders
Set the Row Count

Trap the **GetData** event for informing the InaGrid control of data or use the InaHelper control if you want to prepopulate the InaGrid control

Trap the **OnSelect** event and the **IsSelected** event or use the InaHelper control for selections

How do the InaGrid events flow?

Cell

GetData - Cell value requested

GetFormat - Cell format requested (this overrides all default formatting, such as BackColor, TextColor, HighLightColor, etc)

OnDraw - Owner Draw Cell paint requested

Selection

IsSelected - Row and/or Column selected request

OnSelect - inform Row and/or Column selected

Edit

OnEditCell - inform Cell is about to be edited and request Cell edit object or cancel Cell edit

GetEditLic - Cell edit object lic value requested

OnInitEditCell - inform Cell edit object has been created and request modifications (negative values mean InaGrid

positions Cell editobject)

SetData - inform value of Cell edit object, when changed and edit object loses focus

Columns

OnResizeColumn - inform Column has been resized

OnMoveColumn - inform Column has been moved

Drag Drop

Dragging

OLEStartDrag

OLEGiveFeedback

OLECompleteDrag

Drag Drop

OLEDragOver Begin

OLEDragOver

OLEDragOver End or **OLEDragDrop**

When using InaHelper, all InaGrid events are still valid and are fired after InaHelper has returned the event.

If you use InaHelper for Row or Cell selections then it may not be necessary to trap the selection events.

If you use InaHelper for Cell population then it may not be necessary to trap the **GetData** event

Mouse events - use GetRowNumber(y) and GetColumnNumber(x) to retrieve the Cell position

Where can negative or positive numbers be used (Special Properties)?

OnInitEditCell event - to position cell edit object. Values specify the extra width and height of the cell edit window. Negative values allow the InaGrid control to position the object using these values. The default is positive (InaGrid does not move the edit cell).

RowHeight property - to adjust the row height. A negative value switches back to auto row height (the default).

ResizeRows property - to allow the user to resize rows through a row cursor in the number column (ShowNumbers property). A positive number specifies that minimum row height that can be adjusted by the cursor. A negative number will cause the row height to default when it falls below the value. The default is false - no cursor.

How do I stop the grid refreshing while scrolling?

ScrollOnThumbTrack property

Can I make the grid navigate like Microsoft Excel?

EditTabNavigate property

What kind of licensing agreement do I need to distribute InaGrid control?

The distribution of the InaGrid control is controlled by a license agreement. As a developer using the InaGrid control, you are required to purchase a license in order to distribute applications that contain one or more InaGrid controls. Without this license, the InaGrid control is considered under evaluation and displays a message box to indicate this. When you purchase a license, you are given a license file (*.lic) that must be copied to the directory that the InaGrid control (*InaGrid.ocx*) resides in. You must not redistribute the license file.

If you are using Visual C++, you may define a variable that is passed as a parameter to the Create function. The contents of this variable contain the first line of text from the license file. Use the following method for defining and passing the license variable:

In Visual C++ 5.0 and above:

```
static const CString strLicenseKey = "first line of text from license file";
BSTR bstrLicenseKey = strLicenseKey.AllocSysString();
BOOL bRet = m_wndGrid.Create(NULL, NULL, CRect(0, 0, 100, 100), this, 1, NULL, FALSE, bstrLicenseKey);
SysFreeString(bstrLicenseKey);
```

Is there a way of selecting the whole row?

InaGrid provides a Row Select mode. Row selections can still be achieved in cell select mode (normally implemented by a left click in the Row Number area).

When InaGrid needs to highlight data (a selection was made), it uses an event called IsSelected. Like most InaGrid events, IsSelected passes the row, column and a flag for you to set that states whether this cell should be highlighted or not.

When I run the MFC samples I get a "Failed to create empty document" message and the application exits. Why?

InaGrid is not registered correctly. If you are evaluating InaGrid then you need to download the [InaGrid OCX Setup](#) and install the InaGrid control.

GetData in an MFC application fails to add the OLE_COLOR pColor parameter correctly?

Make sure the lines in the .cpp file read:

```
ON_EVENT(object, 1, 1 /* OnGetData */, OnGetData, VTS_CY VTS_DISPATCH VTS_PBSTR VTS_PCOLOR)
```

And

```
void object::OnGetData(CURRENCY nRow, LPDISPATCH pColumn, BSTR FAR* pValue, OLE_COLOR* pColor)
```

And in .h file:

```
afx_msg void OnGetData(CURRENCY nRow, LPDISPATCH pColumn, BSTR FAR* pValue, OLE_COLOR* pColor);
```

Building some MFC samples produces these errors?

```
C:\VC\Studio\VC\INCLUDE\xutility(45) : error C2678: binary '<' : no operator defined which takes a left-hand operand of type 'const struct SColumnInfo' (or there is no acceptable conversion)
```

This is a service pack issue. Inagrid works with VC5.0 Service Pack 2 and above.

What is the InaHelper control and may I distribute it?

The InaHelper.dll is an ATL control. It must be registered for it to work just like the InaGrid.ocx control.

The InaHelper control is a simple to use control for InaGrid row and/or column selections (**InaSelect** object) and it may also be used for prepopulating the grid (**InaData** object). The InaHelperDemo sample shows how to use the InaHelper control from all aspects including setting the edit control type for different columns. You may redistribute

the InaHelper control with your InaGrid applications if you are a licensed InaGrid developer. The InaHelper control does not require a license (.lic) file.

(You may not redistribute the InaGrid license file)

What samples are available?

MFC (..\program files\inabyte inc\inagrid\samples\mfc)

AltDragDrop - alternate sample of Drag Drop

DragDrop - Drag Drop using the InaGrid ole events

InaAutoSize - simple use of resizing columns to fit headers or data

InaCharMap - Unicode font viewer and clipboard helper. Uses the InaHelper control for cell selections.

InaCircle - shows owner draw columns. **OnDraw** event.

InaDAO - Access databases.

InaGridDemo - pretty much everything.

InaHelperDemo - demonstrates grid row selection and data population using the InaHelper control.

InaHexView - probably the fastest Hex viewer you'll ever see.

InaSelect - selections using the InaGrid control **OnSelect** and **IsSelected** events.

InaSimpleGrid - the basics of InaGrid.

InaSolver - simple accounting sample.

VB (..\program files\inabyte inc\inagrid\samples\vb)

DragDropVb - Drag Drop using the InaGrid ole events

InaDAOVb - Access databases

InaGridDemoVb - pretty much everything.

InaHexViewVb - probably the second fastest Hex viewer you'll ever see.

InaSimpleGridVb - the basics of InaGrid.

VbCellTest - simple grid using the InaHelper control for cell selections.

VbRowTest - simple grid using the InaHelper control for row selections.

How do I contact Inabyte Inc.?

By Phone 415-883-3407

By Fax 415-898-1652 (Sales and Support)

By Email support@inabyte.com

info@inabyte.com

By Internet <http://www.inabyte.com/support.html>

<http://www.inabyte.com>

By US Mail Inabyte Inc.

5 Betty Lane

Novato, CA 94947

USA

Samples provided

See what InaGrid can do for yourself. [Download](#) InaGrid and/or any of its sample test applications (MFC and VB) and see some of its capabilities.