

# First Hand Report About dBASE Plus Development

Presented by  
Martin Kay  
CTO, dataBased Intelligence, Inc.

at

***dB D-A-C-H 2007***  
November 9 and 10, 2007  
Marienheide, Germany

*The following is a summary of the information presented about dBASE Plus:*

## Part I

dBASE Plus 2.61.4 is under development and will likely be released by the end of 2007.

Work has been continuing on the Product Roadmap plan with progress on implementing ADO and in porting dBASE to the newest Microsoft C++ compiler. There is still much work to complete.

A demonstration of the latest dBASE Plus beta build was presented showing the following enhancements:

When designing a Grid in the Form Designer, if custom columns are defined, you can now:

- size columns, move columns, and set the grid's *cellHeight* (row height) by using the mouse
- select a column's *editorControl* or *headingControl* into the Inspector by left clicking them with the mouse.

Grid level mouse event handlers will now fire anywhere on a grid as long as the event handler is defined and is not overridden by a matching columnHeading or editorControl event. This includes mouse events on a column header, row header, grid cell, or grid background.

Grid editorControls (*columnEntryfield*, *columnSpinBox*, *columnCheckBox*, *columnComboBox*, *columnEditor*) now have a full set of mouse events:  
*onLeftMouseDown*, *onLeftMouseUp*, *onLeftDbIClick*,  
*onRightMouseDown*, *onRightMouseUp*, *onRightDbIClick*,  
*onMiddleMouseDown*, *onMiddleMouseUp*, *onMiddleDbIClick*,  
*onMouseMove*

If a mouse event is implemented for an editorControl or columnHeading control, it overrides the matching grid level event.

The columnComboBox control has been enhanced to have the following properties and events to match the comboBox control:

*Style* (with options of 0 - DropDown or 1 – DropDownList)  
*AutoTrim*  
*MaxLength*  
*SelectAll*  
*BeforeDropDown*  
*BeforeCloseUp*  
*OnChange*  
*OnChangeCancel*  
*OnChangeCommitted*  
*BeforeEditPaint*  
*OnEditPaint*

In addition, the *columnComboBox* now auto adjusts its height to match the current grid *cellHeight*.

Under consideration is adding mouse events to *gridColumn* objects as well. If this is done, then any *editorControl* or *headingControl* events will override *gridColumn* level events and the the *gridColumn* level events will override any matching grid level events.

Added new Grid property: *alwaysDrawCheckbox*

- The default value for *alwaysDrawCheckbox* is True which causes any *columnCheckbox* controls to be painted with a checkbox for all cells in the column.
- When *alwaysDrawCheckbox* is set to False, the checkbox is only drawn in a cell if it has focus.

## Bug Fixes

*(The following bug fix list was displayed but not reviewed in depth)*

QAID: 6243 - Fixed the conversion for the columnEditor's dropDownHeight from the current form metric to pixels when expanding the columnEditor. This led to the expanded columnEditor being set to a much larger height than it should be when the form's metric was set to 6 (Pixels). Also made additional fixes to correctly convert the columnEditor's default

height, width, and left margin to the current form metric.

QAID: 6319 - Modified columnComboBox so it auto adjusts its height when the grid's cellHeight is changed or the user manually resizes the grid's row height.

QAID: 6299 – Fixed regression that caused incomplete painting of components on a notebook when an XP Visual Style was active and a subform containing a datalinked grid was opened over the form.

QAID: 6300 - Implemented improved design to ensure that anchored components (including grids with default columns) are sized correctly during form open processing. This ensures that the width of default grid columns is set properly.

QAID: 1253 - Updated grid to support passing correct mouse coordinates to column header mouse events. Previously, the column header mouse events always returned zero. Also fixed: - Firing of Grid level mouse events when clicking to the right of the last column header or below the last row header.

QAID: 6151 - Fixed autosizing of default grid column widths when the grid's anchor property is not set to its default value.

QAID: 6272 - Fixed Table Designer Save As routine that was incorrectly attempting to free a string containing the Save As filename which could lead to unpredictable behaviour.

QAID: 6273 - Fixed potential buffer overflow in routine that processes the picture property's S (scroll) option. If the value passed for S is 5 or more digits, then a buffer overflow would occur. This is rare as S would normally be between 1 and 3 digits.

QAID: 6274 - In routine used to validate dates in entryfields, modified code to ensure that a month greater than 12 can never be used to access an array of month lengths beyond the end of the array.

QAID: 6275 - Fixed potential string buffer overrun when setting up icon labels for filenames in Navigator.

QAID: 6277 - Ensured a NULL pointer is not used in Undo function in Report Designer

QAID: 6278 - Fix potential memory leak when opening a database when a parameter is found that is of an unknown type.

QAID: 6279 - Fixed small memory leak in the following XDMML commands:  
INDEX ON, COPY INDEXES, COPY TAG when the index being created or copied is a Primary key index.

QAID: 6280 - Modified FrameWindow's open sequence to work more efficiently. This will be faster and save stack space.

QAID: 6281 - Ensured that a null pointer cannot be used when opening the Count dialog.

QAID: 6282 - Added test to ensure that a Null pointer cannot be used when double clicking on a table in the Navigator or when completing the Form Wizard. There was a chance that a Null pointer could be referenced if the Form creation failed.

QAID: 6289 - Fixed implementation of ActiveX component to properly release memory allocated to hold the license key's string in the event that the call to retrieve the license key string fails.

QAID: 6290 - Fixed memory leak in ActiveX object that occurred if the call to retrieve online help information failed.

QAID: 6291 - Improved error testing when searching for correct resource .dll (ex: Plus\_en.dll) to load during dBASE startup.

QAID: 6292 - Modified ActiveX object to ensure that an ActiveX's license string is passed with the correct type to the constructor of the ActiveX Container object.

QAID: 6293 - Corrected call to release memory for a temporary array used during processing of a DDETopic onExecute event.

QAID: 6294 - Added error checks to date object's ToGMTString() and ToString() methods

QAID: 642 – Fixed firing of onLeftDbfClick event for GRID class

QAID: 1396 - Modified Grid to support firing of mouse events when clicking on a grid cell when rowselect is false.

Also fixes the following QAIDs which are special cases of this problem:

642 - Grid Class onLeftDbfClick Fails to Fire

1995 - onLeftMouseDown only fires once for any row, unless you move off it and back on it

5670 - onRightMouseUp appears not to fire in Grid when rowselect = false

1995 - onLeftMouseDown only fires once for any row, unless you move off it and back on it

QAID: 5670 – Fixed OnRightMouseUp appears not to fire in Grid when rowSelect = false

QAID: 6239 - Fixed regression where components on a notebook are not visible when a form containing a notebook is first opened when the notebook has no border and no manifest file is used (so old common controls are being used).

QAID: 6259 - Fixed regression that caused a Line object (or Shape object) on a Notebook to be clipped by another object when that other object's visible property was set to False.

QAID: 6268 - Modified container class so it repaints its controls when container is a child of a notebook object.

QAID: 6269 - Fixed firing of combobox.key() and combobox.onKey() events when user presses the RETURN key (for style 0 and 1 comboboxes).

QAID: 6276 - Fixed potential use of a NULL pointer and prevented scanning past end of string buffer in reporting records copied during COPY command and other commands that report progress

## **Part II - dBASE Plus 3.x**

Demonstrated how new ADO Database Aliases are defined using enhanced Properties dialog.

Demonstrated connecting via ADO to a database (using MS SQL Express) and seeing the tables displayed in the Navigator.

Demonstrated double clicking on a table in the Navigator to view its contents in an auto-generated form and grid.

Demonstrated running several forms that use ADODatabase, ADOQuery, and ADORowset components

Showed source code of ADO based form to show how ADO components are instantiated instead of database, query, and rowset objects.

Discussed how ADO development was still underway and that code was not yet ready for release. It would likely be 9 more months before ADO is ready.

Elaborated on Product Roadmap by stating that the roadmap item: "Maximize XP and Vista compatibility" will include:

- Code signing dBASE installers and executables to minimize security issues.
- Modifying dBASE to support code signing of exe's built using dBASE.

## Part III

### Future Directions - What comes after current Roadmap?

Discussed the benefits of each of the following items:

- dBASE Runtime built as a DLL
- Embed dBASE Code in web pages - like Perl or ASP
- Create dbf engine separate from BDE.
- Improve execution speed of dBASE Virtual Machine

Solicited input from conference attendees on whether or not they are interested in any of the following:

- Upgraded dbf format?
- Unbundle dBASE Compiler from IDE?
- Add 64 bit support?
- Multithreading Virtual Machine?
- Ability to use multiple CPU's?
- dBASE .Net?
- dBASE on Linux?
- Open Source dBASE?