Drag and Drop Data between 4D and Excel

By Atanas Atanassov, Technical Services Team Member, 4D Inc.

Technical Note 08-19

Abstract

This Technical Note covers how to implement the new Drag and Drop mechanism as well as the pasteboard functionality in 4D v11 SQL to copy text data between a 4D Listbox and Excel spreadsheet.

Introduction

As a developer, it will be great to give your customer a way to easily migrate data between a Microsoft Excel Spreadsheet and 4D. Why is this so important? Microsoft Excel is a broadly used application and many end users are familiar with its functions. Providing the ability to move data between the two applications can increase an end user's productivity. To help implement such a feature the new 4D drag and drop mechanism available in 4D v11 SQL is used. This new mechanism along with the new pasteboard feature allows copying data between an Excel spreadsheet and a 4D Listbox.

Example Database

For a more complete understanding of the material, an example database is included. In order to use the example database, the Microsoft Excel application is required. The following Excel versions that may be used with this Technical Note are:

- Microsoft Excel 2003 (Windows)
- Microsoft Excel 2007 (Windows)
- Microsoft Excel 2008 (MacOs X)

New Features

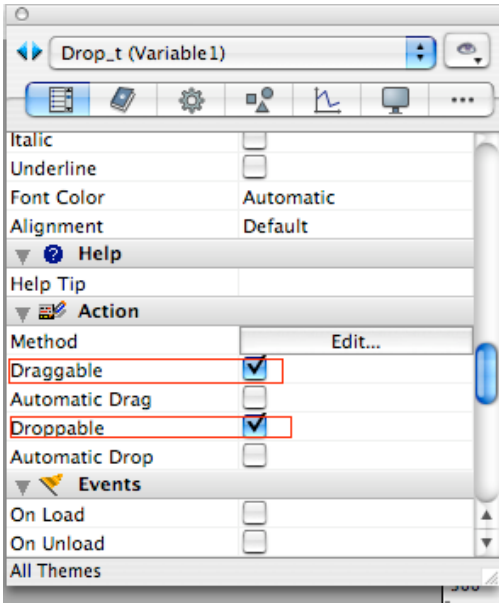
4D v11 SQL introduces a new Drag and Drop mechanism as well as a new pasteboard (originally known as clipboard). These new enhancements provide the developers a means to develop features in their application which were not natively available in previous versions of 4D. In Previous versions a plug-in would be needed to achieve similar behavior.

Drag and Drop mechanism in 4D v11 SQL

Drag and Drop mechanism built in to 4D v11 SQL comes in two flavors, automatic and manual. Both allow moving data from one object to another in the same process or between processes. Also it is possible to move objects between forms in 4D and other applications on your desktop. Lastly, the new On Drop Database

Method allows dropping objects directly onto the application without having a form in the background. With this added database method, it is possible to open a 4D Write document simply by dragging the document onto the 4D application icon.

In order to implement the drag and drop manual functionality for an object, its property needs to have the following enabled:



Italic	☐
Underline	☐
Font Color	Automatic
Alignment	Default
Help Tip	
Method	Edit...
Draggable	☑
Automatic Drag	☐
Droppable	☑
Automatic Drop	☐
On Load	☐
On Unload	☐

***Note**: By default these properties are not checked!*

In the "Action" theme the *Draggable* and *Droppable* properties allows the events On Drag Over and On Drop to trigger. Once enabled, data can either be dragged or dropped onto an object that has these properties.

***Note**: It is important to press and hold the **Alt** (for Windows) or **Option (**for Mac OS) key when dragging and dropping.*

In addition, fields,variables, combo boxes, and listboxes in 4D v11 SQL have the new *Automatic Drag* and *Automatic Drop* properties. When enabled, it allows copying and moving text or pictures between form areas without any code. Below is a list of where text and pictures can be copied or moved from:

- Same 4D area
- Between 2 4D areas
- Between 4D and another application (i.e wordpad)

Drag and Drop form events

On Drop

This event occurs once the mouse button is released over the destination object. You don't need to check for data compatibility, if this has been done during the first phase of dragging the object.

On Drag Over

This event controls the first phase of drag and drop. At this point the data compatibility between the objects is checked and it is triggered when the mouse pointer is moved over the object.

On Begin Drag Over

This event is called in context of the source object and it is generated when the Draggable property is checked for the source object.

Note: *In our sample database, we won't check for type compatibility; since we drag between an Microsoft Excel spreadsheet and a listbox object, we will assume that the type is text only.*

Drag and Drop Commands

DRAG AND DROP PROPERTIES (srcObject; srcElement; srcProcess)

srcObject	Pointer	Pointer to drag-and-drop source object
srcElement	Number	Dragged array element number, or Dragged listbox row number, or Dragged hierarchical list item, or -1 if source object is neither an array nor a list box nor a hierarchical list
srcProcess	Number	Source process number

This function returns information about the source object. Typically, you call it in the context of an object method or a subroutine when the On Drop or On Drop Over events occur for the destination object.

Drop position {(columnNumber)}

columnNumber	Longint	List box column number or -1 if the drop occurs beyond the last column
Function result	Number	Number (array/list box) or • Position (hierarchical list) or • Position in string (text/combo box) of destination item or -1 if drop occurred beyond the last array element or list item

This command is used to find out the destination position of a complex object like an array, a listbox or a text field and it is called in the context of an object method activated from On Drop and On Drag Over events for the destination object.

Pasteboard

Another feature used for this Technical Note is the pasteboard. In previous versions, it was known as the clipboard. 4D v11 SQL introduces new commands for the pasteboard as well as new functionality.

The commands can be used both for managing copy/paste actions (Clipboard management), as well as inter-application drag and drop actions, which this Technical Note exhibits. 4D uses two data pasteboards:

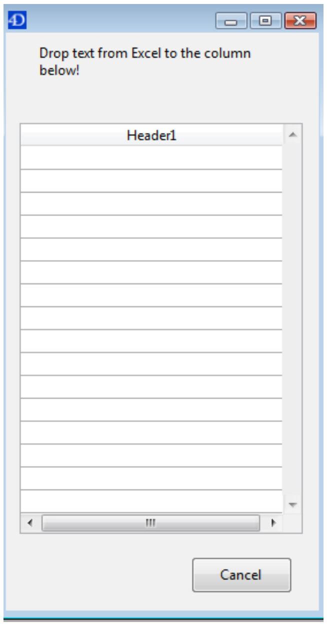
- copied (or cut) data, which is the actual clipboard that was already present in previous versions.
- data being dragged and dropped.

These two pasteboards are managed using the same commands. You access one or the other depending on the context. Drag and Drop is managed from the pasteboard only with the On Drop, On Drag Over, On Begin Drag Over form events and the On Drop Database method. Other than that the Drag and Drop Pasteboard cannot be accessed.
The pasteboard commands used in the sample database are Get text from pasteboard and Set text to pasteboard.

***Note**: We do not need to clear the pasteboard after the drag and drop, because contents of the pasteboard are removed after we drop the contents.*

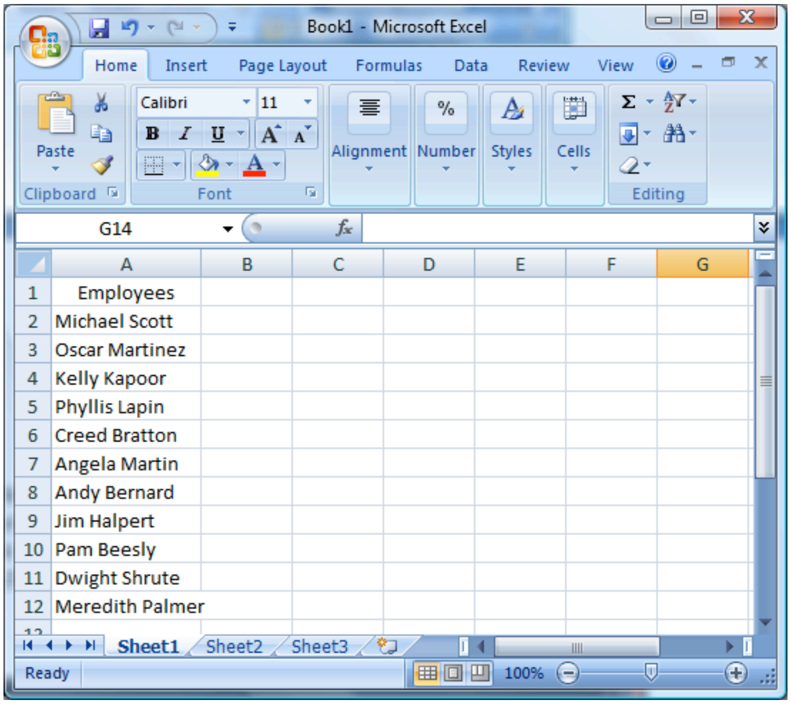
Drag from Microsoft Excel Spreadsheet and Drop to 4D Listbox

When the database opens it starts in the Application environment. From the File menu select the menu item "Open Listbox". The following dialog will then appear.



Drop text from Excel to the column below!

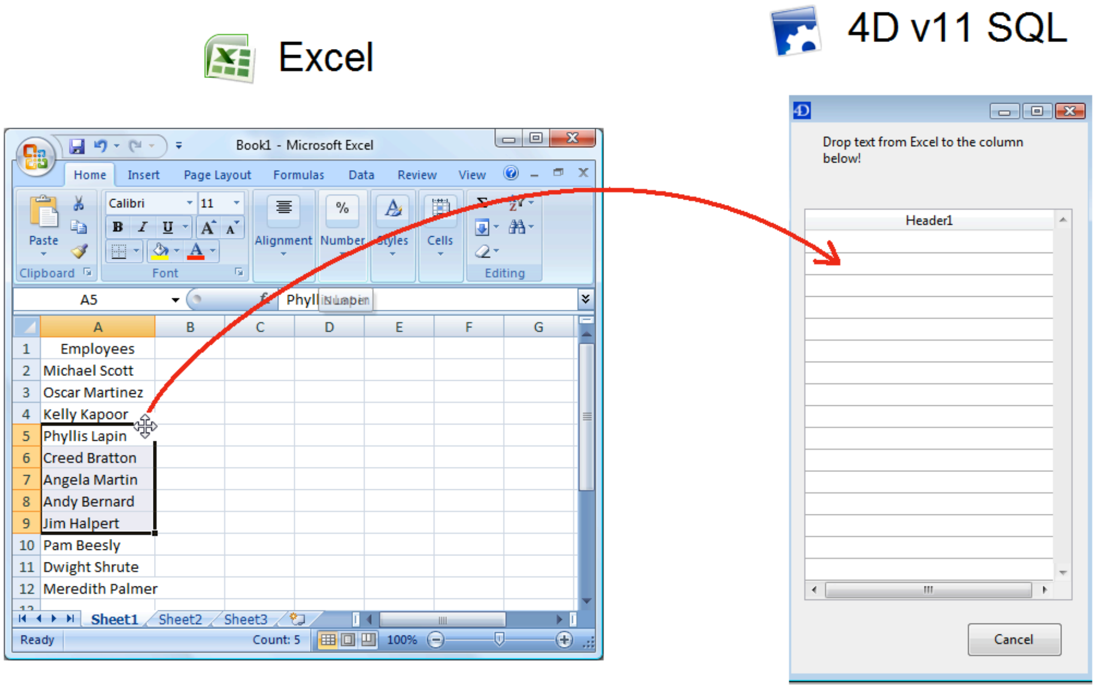
Open a Microsoft Excel file.



	A
1	Employees
2	Michael Scott
3	Oscar Martinez
4	Kelly Kapoor
5	Phyllis Lapin
6	Creed Bratton
7	Angela Martin
8	Andy Bernard
9	Jim Halpert
10	Pam Beesly
11	Dwight Shrute
12	Meredith Palmer

Now select cells from the excel worksheet. Then place the mouse cursor on the outline of the selection until the move cursor appears. Once the move cursor appears drag the selection and drop it onto the listbox column.

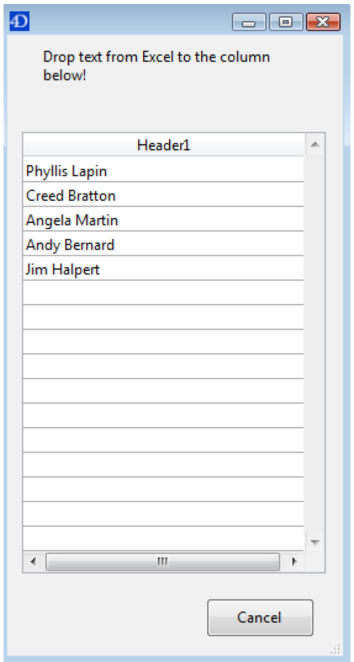
***Note**: The example is restricted to copy only 1 column.*



	A
1	Employees
2	Michael Scott
3	Oscar Martinez
4	Kelly Kapoor
5	Phyllis Lapin
6	Creed Bratton
7	Angela Martin
8	Andy Bernard
9	Jim Halpert
10	Pam Beesly
11	Dwight Shrute
12	Meredith Palmer

Drop text from Excel to the column below!

The result should look like the following:



Drop text from Excel to the column below!

Header1
Phyllis Lapin
Creed Bratton
Angela Martin
Andy Bernard
Jim Halpert

Here is the code that gets triggered during the On Drop event of the column.

```
C_TEXT($Text)  `Text from the Pastebard
C_LONGINT($counter)  `count tab delimiters
Case of
   : (Form event=On Drop )

         $Text:=Get text from pasteboard
         $counter:=M_CountDelimiter ($Text)  `Count delimiters in a row
         M_Tokenizer ($Text)

End case
```

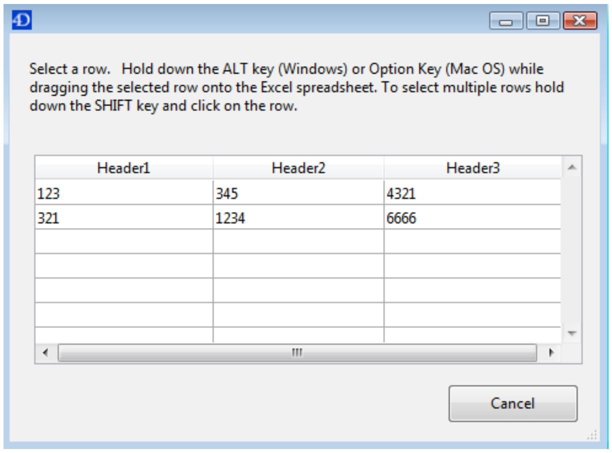
M_CountDelimiter counts how many column delimiters (**Char**(Tab)) are in the text. This project method accepts text from the Pasteboard and the return value is the number of delimiters in each row.

The last method that executes during the call *M_Tokenizer*. This method gets the text from the pasteboard, tokenizes the text and puts each token into the relevant column in the listbox.
The idea behind this method is very simple. We take one row and parse it, and then we delete this row and repeat the same procedure till we get the last row. At this point all tokens from the text are in the listbox.

***Note**: Microsoft Excel for Mac OS, does not have a carriage return character after the last row. To make this method compatible for Mac we need first to check for the position of the carriage return. If we have a carriage return at the end of the row, we take all text till the carriage return as one row, otherwise we take the rest of the text as one row.*

Drag from 4D Listbox and Drop to Microsoft Excel Spreadsheet

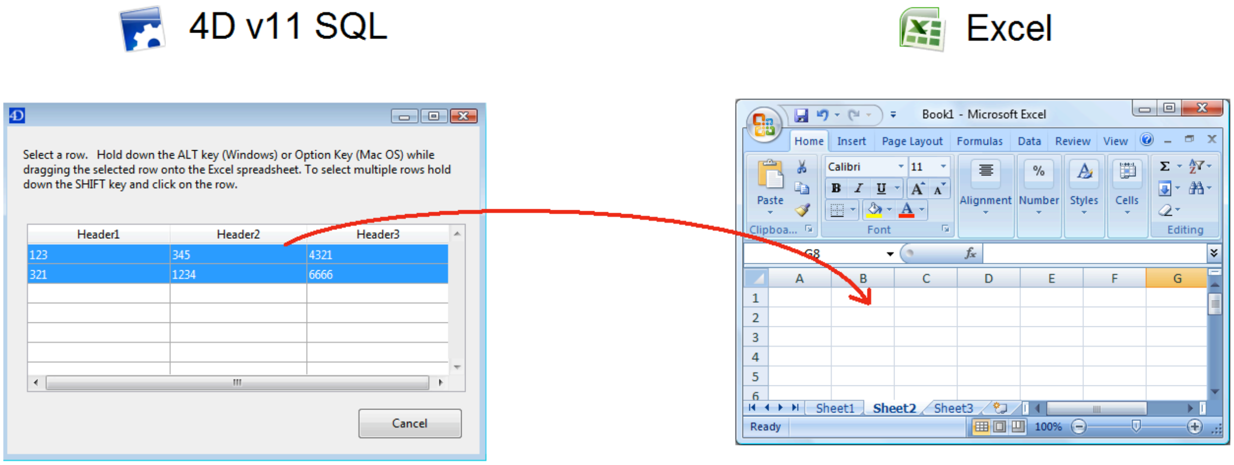
Selecting the menu item "Open form to Drag from" from the File menu will display the following dialog:



Select a row. Hold down the ALT key (Windows) or Option Key (Mac OS) while dragging the selected row onto the Excel spreadsheet. To select multiple rows hold down the SHIFT key and click on the row.

Header1	Header2	Header3
123	345	4321
321	1234	6666

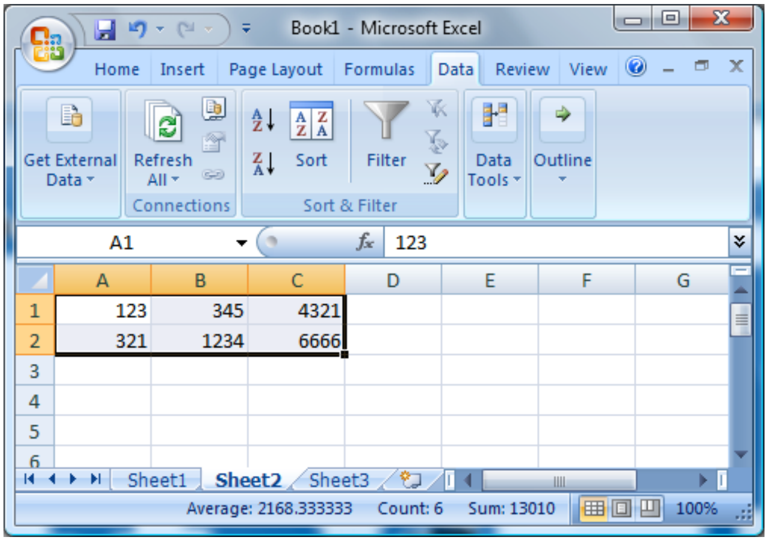
Select the two rows. While holding down the **Alt** (for Windows) or **Option (**for Mac OS), drag the selection and drop it onto the Microsoft Excel worksheet.



Select a row. Hold down the ALT key (Windows) or Option Key (Mac OS) while dragging the selected row onto the Excel spreadsheet. To select multiple rows hold down the SHIFT key and click on the row.

Header1	Header2	Header3
123	345	4321
321	1234	6666

The end result is the following:



A1 = 123

	A	B	C	D	E	F	G
1	123	345	4321				
2	321	1234	6666				
3							
4							
5							
6							

Average: 2168.333333 Count: 6 Sum: 13010

The Project form “droptext” has a form method, which creates and initializes the listbox columns and calls a project method named *droptext.*
Below is the code for the droptext method.

```
C_TEXT($Text_t)
C_TEXT($ntext_t)
C_LONGINT($col_i;$row_i;$index;$index_col)
C_POINTER($colVar_ptr)

$col_i:=Get number of listbox columns(*;"List Box1")
$row_i:=Get number of listbox rows(*;"List Box1")


For ($index;1;$row_i)
  If (List Box1{$index}=True)
    For ($index_col;1;$col_i)
```

```
      If ($index_col<$col_i)  `Check if this is the last element in current row
        $colVar_ptr:=Get pointer("Column"+String($index_col))
        $Text_t:=$colVar_ptr->{$index}+Char(Tab )
        $ntext_t:=$ntext_t+$Text_t
      Else   `if this is the last element in the row does't add delimiter
        $colVar_ptr:=Get pointer("Column"+String($index_col))
        $Text_t:=$colVar_ptr->{$index}
        $ntext_t:=$ntext_t+$Text_t
      End if
    End for
    $ntext_t:=$ntext_t+Char(Carriage return )
    SET TEXT TO PASTEBOARD($ntext_t)
   End if
End for
```

The method starts off by obtaining the amount of columns and rows in the listbox. Then the code goes on to check what rows have been selected. This is achieved by using the listbox Boolean array; the row index of the listbox is represented as an array of Booleans and if the row is selected, its index value in the array is True.

This allows selection of either one row or a selection of rows. Next, the selected row or rows are appropriately formatted by including text delimiters and carriage returns. The formatted text then is set to the pasteboard by using the command *SET TEXT TO PASTEBOARD*.

Conclusion

This Technical Note has shown how the Drag and Drop mechanism and pasteboard features in 4D v11 SQL can be used to easily copy text data between Excel and 4D. Developers can use the small example provided and expand on it to suit their customer's needs. Adding such a feature to an already existing production database can boost the productivity of end users who extensively use the Excel spreadsheet software.

Related Sources

Technical Note reference#48915- 4D v11 SQL Pasteboard