Quicklist - Fast lists for FileMaker

In FileMaker you use value lists for various things. Those lists are simply text separated with new line characters. You can build them easily in FileMaker calculations by appending new text to a list and using the ¶ character. To query a list you can use ValueCount and GetValue functions.

Performance

If you start to process lists with 10000 or more items, you will see degrading performance. Appending list entries requires FileMaker to make a new text and allocate memory for new size, copy existing text and new text together into the new text and return it. The longer the list becomes, the longer the copy process will take. For accessing list in a loop, you call GetValue a lot. Each time you query a value, the FileMaker runtime has to loop over the content of the text and find the nth item. The longer the text is, the longer this search will take.

Query values

The MBS Plugin now offers QuickList functions. With QuickList.New function you create a new list passing the text of existing list. The plugin will parse it once for fast access. You query the number of items using QuickList.Count function. The function QuickList.GetValue now reads a value and if you do a loop over thousands of entries. You will notice that for a lot of entries, the plugin will be magnitudes faster. When done, please free the list from memory using QuickList.Free function.

Building lists

To build a list, you also call QuickList.New. You can pass of course a text with initial list items. Than use QuickList.AddValue to add a value as often as you need. To optimize memory you can reserve memory upfront for a given number of entries using the QuickList.Reserve function. Now we you added enough entries to the list, you can use QuickList.GetList to get the whole list as text. Finally you release the memory of the list.

Results

The client we developed this functions had to import a text file with 150,000 lines and create new records. Just by using QuickList.GetValue instead of GetValue, the time needed for importing went down from over 20 minutes down to just below 3 minutes. Further optimization gained more speed ups like also using QuickList for parsing each row. While

QuickList is built for lists with ¶, but the QuickList.New and QuickList.SetList functions take an optional parameter for delimiter. This way you can also use tab character (pass 9) or "|" (vertical line) as delimiter.