



BEYOND THE GRID: SMARTER SORTING & DYNAMIC FILTERING

PETER BRAGG

DATA MATTERS

DATA MATTERS MORE THAN JUST A LIST

- Displaying data is only just the starting point.
- Operators are rarely interested in *all* the records.
- A useful list helps the operator find, compare and select the records that matter.
- That usually means:
 - **Sorting** – ordering the records in a useful way.
 - **Searching** – finding records that match a specific value.
 - **Filtering** – reducing the list to a relevant working set.

DATA MATTERS

SETTING THE BOUNDARIES

- Third-party grid components may provide their own solutions.
- This session focuses on the standard DataFlex cWebList (DataFlex 25.0).
- Data Aware.
- One option would be to populate the list with all records.
- But I won't doing that because:
 - I cannot guarantee performance as the data grows,
 - I want better control over the sort rules.
- Microsoft SQL Server database.
- SQL gives us more options for sorting (and searching) and filtering.

DATA MATTERS

THE WEBLIST RULES

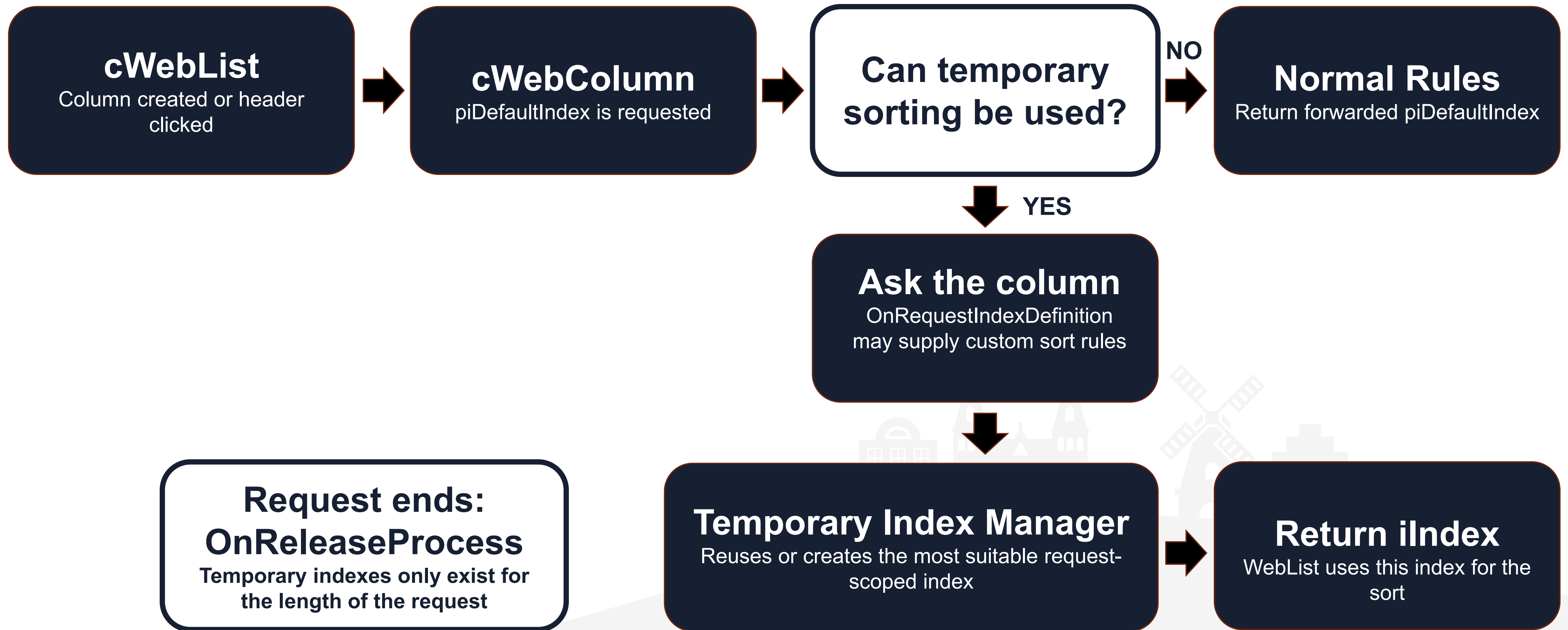
Before we improve the experience, we need to understand what a data-aware cWebList gives us by default:

- It does not automatically contain all records.
- A column can only be sorted if its field (entry item) is part of an index.
- If that field is not the first segment of the index, the sort order may be be what the user expects.
- You cannot sort the list by a foreign field.
- A column can only be searched if it can be sorted.
- Filtering is supported, but is not exposed to the end user by default.

We will address what we can with a reusable **Sorting and Filtering Library**.

SORTING AND FILTERING LIBRARY

Sorting framework: request flow



Important distinction: the column expresses sort intent; the manager handles index reuse/creation

Temporary Index Manager: decision logic

Singleton
Temporary Index Manager
lives within oWebApp

Request Temporary Index
Table + entry field + optional definition



Manager checks
Does a suitable field index already exist,
and is the entry field the first segment?

SUITABLE EXISTING INDEX

FIELD INDEX EXISTS, BUT NOT FIRST

NO SUITABLE INDEX

Use existing index
Field already has an index and is the first
segment

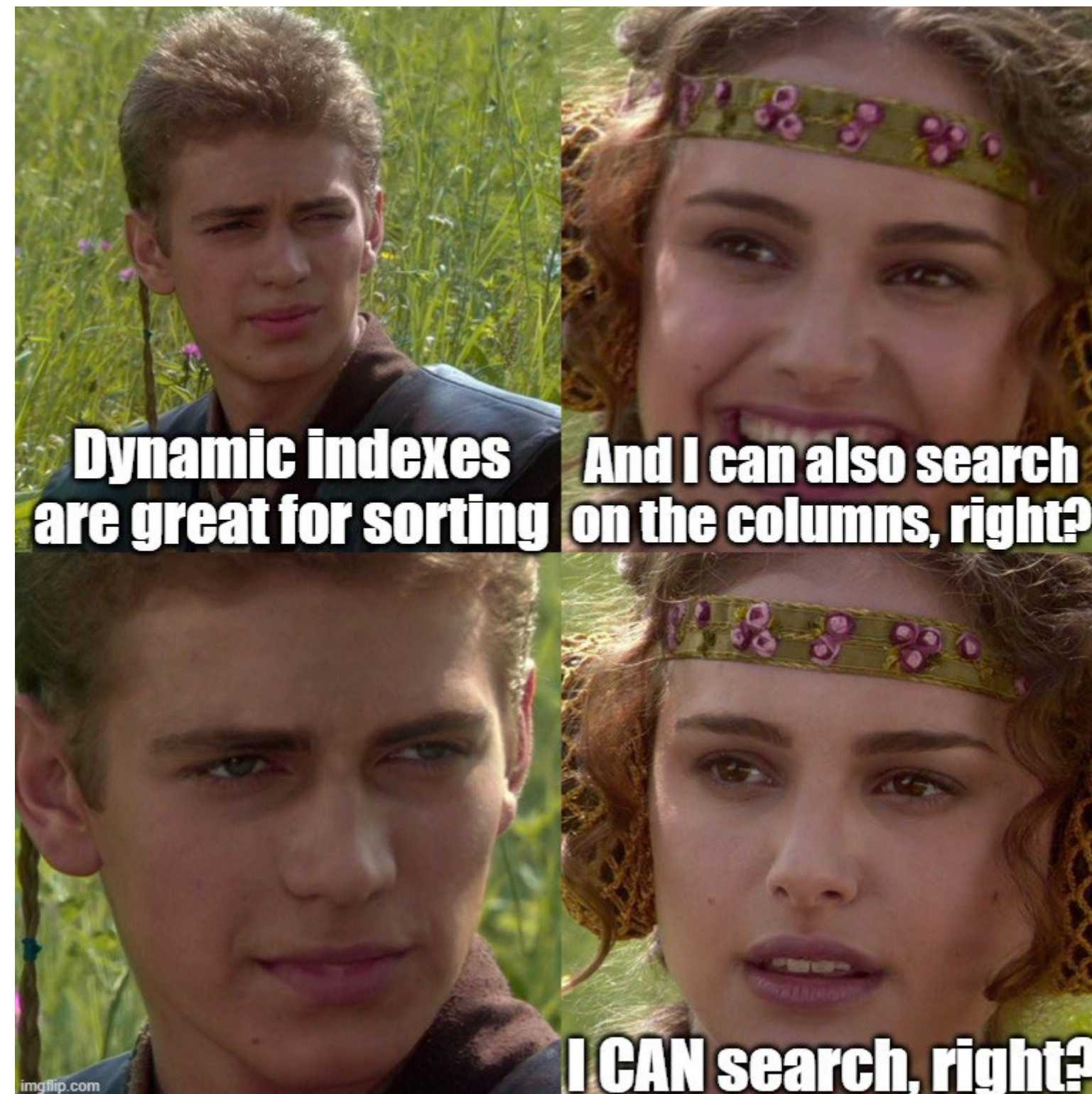
Promote the entry field
Same segments, but with the entry field
first

Create new unique index
Entry field first, then primary-index fields to
make the result unique

Return iIndex
Back the column / WebList

The manager uses a suitable existing index where possible; otherwise, it creates a request scoped index

SORTING AND FILTERING LIBRARY IMPROVED SEARCHING



SORTING AND FILTERING LIBRARY

IMPROVED SEARCHING

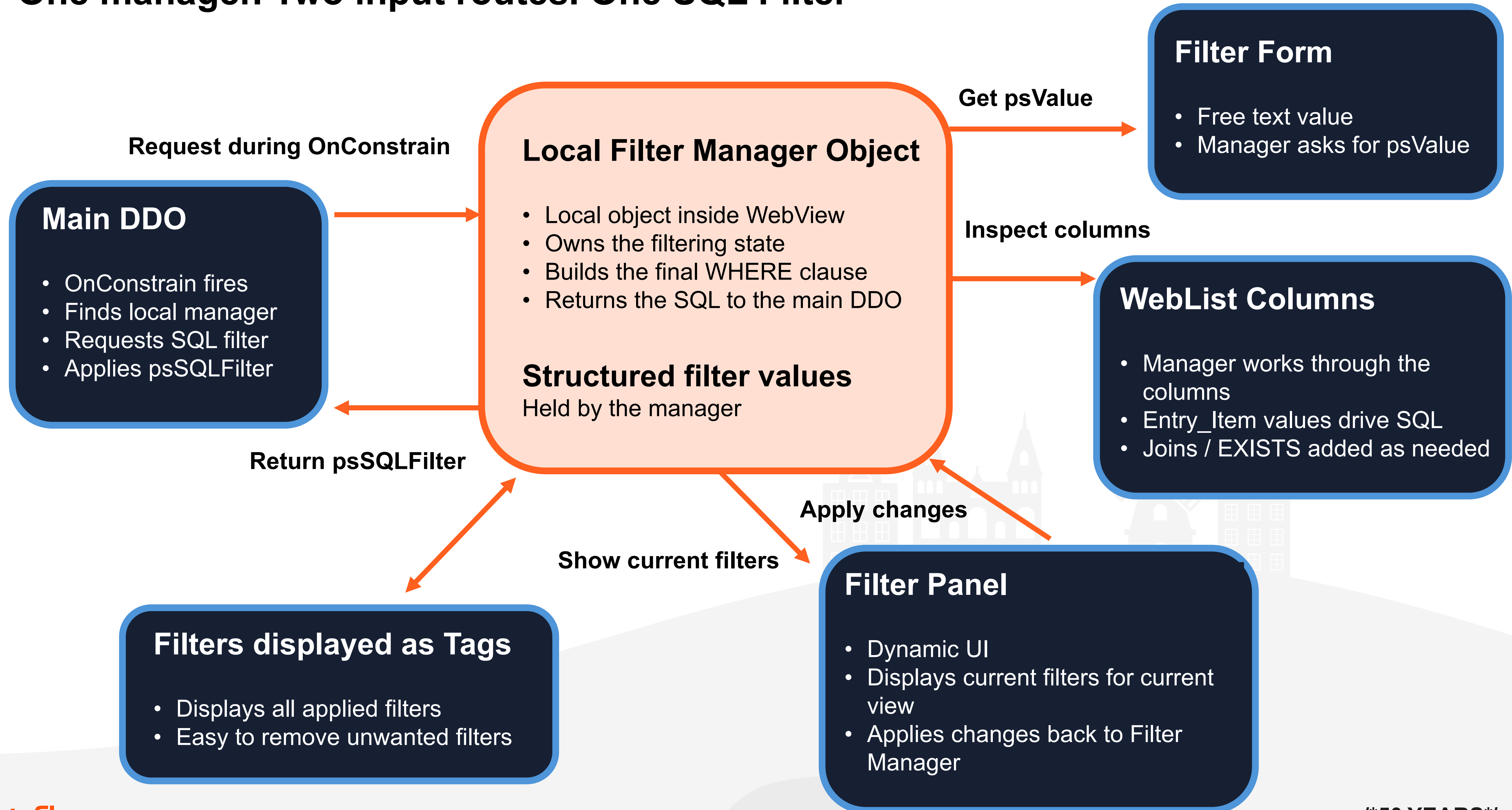
We need to augment RequestColumnLookup:

```
Procedure RequestColumnLookup String sSearchText
  Integer iOldOrdering iNewOrdering
  Handle hoServer

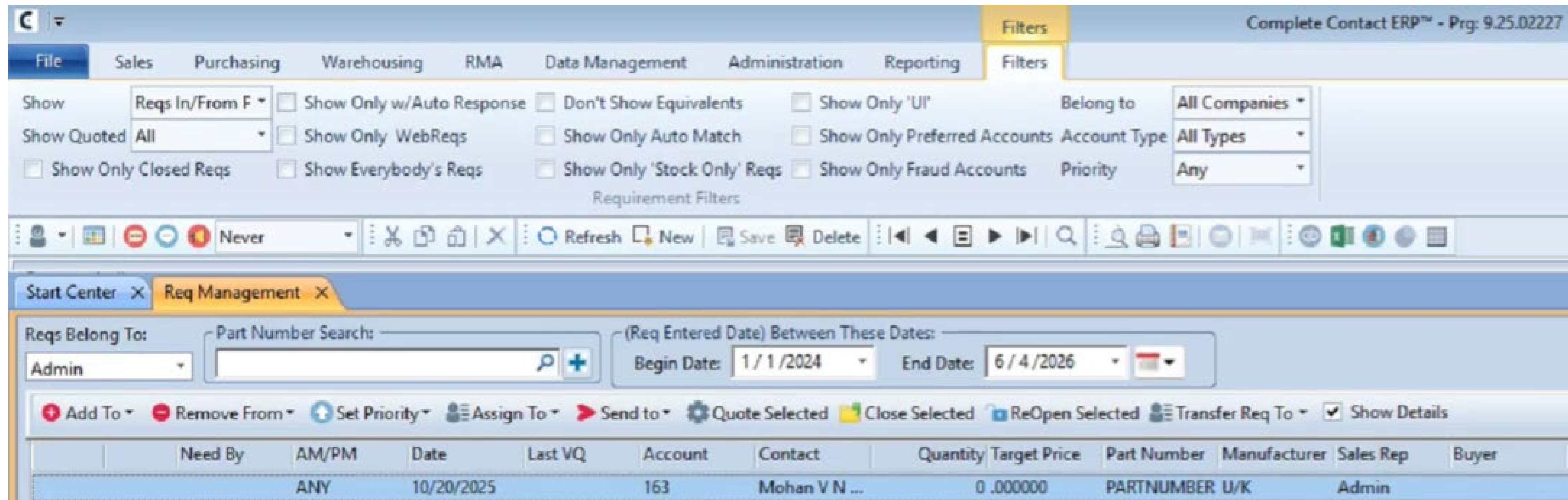
  Delegate Get Server to hoServer
  If (hoServer>0) Begin
    Get Ordering of hoServer to iOldOrdering
    Get piDefaultIndex to iNewOrdering
    Set Ordering of hoServer to iNewOrdering
  End
  Forward Send RequestColumnLookup sSearchText
  If (hoServer>0) Begin
    Set Ordering of hoServer to iOldOrdering
  End
End_Procedure
```

- RequestColumnLookup does not call piDefaultIndex by design.
- Making it do so ensures a suitable index is established.
- We temporarily Set Ordering before restoring it again.

One manager. Two input routes. One SQL Filter



SORTING AND FILTERING LIBRARY ORIGINS



- Dynamic filters in RibbonBar. Additional filters inside view.
- Equivalent in WebApp would take up lots of screen space.
- Wanted to keep the visibility of filters that were currently being applied.

SORTING AND FILTERING LIBRARY ORIGINS

The screenshot displays the CERP (Complete Contact ERP) interface for managing Requirements. The top navigation bar includes the CERP logo, the current page title 'Requirements', and user information 'Scope Admin'. A sidebar on the left lists various modules: Sales, Back, Quick Quote, Contacts, Prospects, Requirements (highlighted), Quotes, Orders, and Invoices.

The main content area features a 'Requirements' table with columns: Need By, AM/PM, Date Entered, Last VQ, Account, Contact, Quantity, Part Number, Manufacturer, Sales Rep, and Internal. The table contains 10 rows of data. Above the table is a toolbar with various actions: Filter Rows, Filters, Load, Add, Edit, Quote, Add To..., Remove..., Assign, Send..., Priority, Close, Re-open, and Edit Related Quote. A 'Filter Rows' dropdown menu is highlighted with an orange box.

Below the table is a summary bar showing: 1 Supplier Quotes, 24 Offers, 4 Notes, 0 Attachments, 0 Auto Match, and 5 Franchise. At the bottom, there are three panels: 'Vendor Quotes' (showing a quote from LAD INDUSTRIES for 1000 units at USD 15.00), 'Offers' (showing offers from LAD INDUSTRIES), and 'Notes' (showing a note about audit table recording).

SORTING AND FILTERING LIBRARY ORIGINS

The screenshot displays the CERP Requirements module interface. A filtering menu is open, showing various filters for requirements. The main table lists requirements with columns for Last VQ, Account, Contact, Quantity, Part Number, Manufacturer, Sales Rep, and Internal. Below the table, there are sections for Vendor Quotes, Offers, and Notes.

Last VQ	Account	Contact	Quantity	Part Number	Manufacturer	Sales Rep	Internal
02/23/2026	566	Ian Scott	8000	12067C101KAT: KEMT		Admin	
02/23/2026	566	Ian Scott	100	PB2109-70	VFI	Mukesh	
02/23/2026	1005	Checking	5000	AD822	AD	Admin	
	1005	Checking	5000	AD822AR-10		Admin	
	1005	Checking	5000	TX822R	AD	Admin	
	163	Mohan V N K Prasad	0	PARTNUMBER	U/K	Mukesh	
	163	Mohan V N K Prasad	2000	LMX2306TMX	U/K	Mukesh	
	163	Mohan V N K Prasad	1025	TK83361MTL	U/K	Mukesh	
	163	Mohan V N K Prasad	0	???HELLO??	U/K	Mukesh	
	163	Mohan V N K Prasad	3000	????	U/K	Mukesh	

SORTING AND FILTERING LIBRARY ORIGINS

The screenshot displays the CERP Requirements module interface. At the top, the 'Requirements' header is visible, along with user information 'Scope Admin'. A navigation sidebar on the left includes options like Home, Dashboard, Sales, Purchasing, Accounting, Warehouse, RMAs, Data, and Automation. The main content area features a table of requirements with columns for 'Need By', 'AM/PM', 'Date Entered', 'Last VQ', 'Account', 'Contact', 'Quantity', 'Part Number', 'Manufacturer', 'Sales Rep', and 'Internal'. A filter bar above the table is highlighted with an orange box, containing 'Filter Rows', 'Filters', 'Load', '+ Add', 'Edit', 'Quote', and date filters: 'Show Quoted: Yes', 'Date From: 01/01/2020', and 'Date To: 12/06/2026'. Below the table, a summary bar shows '1 Supplier Quotes', '24 Offers', '4 Notes', '0 Attachments', '0 Auto Match', and '5 Franchise'. At the bottom, three panels show 'Vendor Quotes', 'Offers', and 'Notes' with search timelines and recent entries.

	Need By	AM/PM	Date Entered	Last VQ	Account	Contact	Quantity	Part Number	Manufacturer	Sales Rep	Internal
<input type="checkbox"/>	ANY		01/14/2025		566	Ian Scott	8000	12067C101KAT: KEMT		Admin	
<input type="checkbox"/>	ANY		11/05/2025	02/23/2026	566	Ian Scott	100	PB2109-70	VF1	Mukesh	
<input checked="" type="checkbox"/>	ANY		10/30/2025	02/23/2026	1005	Checking	5000	AD822	AD	Admin	
<input type="checkbox"/>	ANY		10/30/2025		1005	Checking	5000	AD822AR-10	AD	Admin	
<input type="checkbox"/>	ANY		10/30/2025		1005	Checking	5000	TX822R	AD	Admin	
<input type="checkbox"/>	ANY		10/20/2025		163	Mohan V N K Prasad	3000	7777	U/K	Mukesh	
<input type="checkbox"/>	ANY		08/29/2022	12/23/2025	1005	Checking	0	AD822AR	AD	Admin	
<input type="checkbox"/>	ANY		08/29/2022		1005	Checking	0	SN74LS	MOT	Admin	
<input type="checkbox"/>	ANY		08/03/2022	12/22/2025	1125	Erica Foland	0	SN74LS	MOT	Admin	
<input type="checkbox"/>	ANY		07/01/2022	12/15/2025	ABC Electronics Limite	Brian	1000	AD822AR	AD	Sattidass	

SORTING AND FILTERING LIBRARY

QUICK RECAP – SORTING AND SEARCHING

- Drag/Drop **TemporaryIndexManager** into **oWebApp**.
- Use **OnReleaseProcess** to call **TearDown** to release dynamic indexes.
- Import **mixin_TemporarySortIndexes** to your WebList Column Subclass (and **Send Define_TemporarySortIndexes_Properties**)
- Optionally use **OnRequestIndexDefinition** (column object) to define sort rules.

SORTING AND FILTERING LIBRARY

QUICK RECAP – FILTERING

- Drag/Drop **cWebViewFiltersPanel** into WebView (below Main_DD setting)
- Make sure **Property Handle phoFilterManager** is defined (view subclass?).
 - “Loose Filtering”.
 - Free text filtering. Values split by space unless quoted.
 - WebListFilterForm object has properties for:
 - **pbFilterOnHiddenColumns** (False)
 - **piMinSearchTextLength** (2)
 - **OnSetFilterField** can be used on columns with no Entry_Item.
 - Applied filter values display as Tags.
 - Removing Tag removes it as a filter.

SORTING AND FILTERING LIBRARY

QUICK RECAP – FILTERING

- **OnRegisterFilters** provides structured filters:
 - Specific columns
 - Specific behaviour (efLE, efLT, efEQ, efGE, efGT, efNE, efBegins, efEnds, efContains)
 - Add to table DDO
 - Filter Panel is used to enter and amend values.
 - All dynamic.
 - Save, Apply, Clear.
 - Applied filter values display as Tags.
 - Removing Tag removes it as a filter.

SORTING AND FILTERING LIBRARY

QUICK RECAP – FILTERING

- **Constraints** passed as **Variant[][]**
 - Maps nicely to Validation Tables (e.g. Get TableData to vaConstraints)
 - Can be **Static** or **Dynamic** (pass empty array for dynamic)
 - Use **OnBuildDynamicConstraints** to define constraints at runtime.
 - Can be used for:
 - cWebCombo (use both dimensions)
 - cWebCheckbox (use one dimension)
 - **DefineFilterConstraint** and **GetFilterConstraint** helper functions

```
Variant[][] aConstraints  
Send DefineFilterConstraint (&aConstraints) "" "- Select -"
```

```
Variant[][] aConstraints  
Variant[] aEmpty  
Get FilterConstraint "" "- Select -" to aEmpty  
Move (InsertInArray(aConstraints,0,aEmpty)) to aConstraints
```

SORTING AND FILTERING LIBRARY

FILTERING - MISCELLANEOUS

- Filters support default values:

- Design Time

```
Send RegisterFilter (&aFilters) "First Name" (RefTable(Person.FirstName)) efEQ aConstraints False "John"
```

- Runtime

```
{ MethodType = Event Visibility = Public }  
Procedure OnDefineFilterDefault Integer iColumn Integer eCondition String ByRef sDefault  
End_Procedure
```

SORTING AND FILTERING LIBRARY

FILTERING - MISCELLANEOUS

- Full Control:
 - **OnBuildSqlFilter** event allows developer to control exactly how filter should be applied.

```
// called before filters are built. Use this event when our filters isn't a simple 'table.column' {some condition} 'value'  
{ MethodType = Event Visibility = Public }  
Procedure OnBuildSqlFilter tWebFilter tFilter String ByRef sWhere  
    If (tFilter.iColumn=RefTable(SalesOrderHeader.Status)) Begin  
        If (tFilter.sValue="Open") Begin  
            Move @SQL""  
            SalesOrderHeader.Status IN ('1','2') AND SalesOrder.SalesPersonId !=0 AND "OrderDate" > '17530101'  
            "" to sWhere  
        End  
    End  
End_Procedure
```

SORTING AND FILTERING LIBRARY

FILTERING - MISCELLANEOUS

- Saving and Loading Filters:
 - SaveFilters
 - LoadFilters
 - HasSavedFilters
- Out-the-box:
 - Saves to (and loads from) local storage. You may want to augment these methods and change this location.
 - Allows for only 1 saved set per View/WebList. You may wish to develop on this and allow more.

SORTING AND FILTERING LIBRARY

SUMMARY OF CLASSES

- **Sorting/Searching:**
 - **cTemporaryIndexManager**
 - **mixin_TemporarySortIndexes**
- **Filtering:**
 - **Mixin_DDWebFilters**
 - **cWebViewFiltersPanel**
 - **csWebListFilterForm**
 - **csWebFilterMenuItem** and **csWebLoadFiltersMenuItem**
 - **csWebFilterTagsDisplay**
 - **csWebFilterPanel**

SORTING AND FILTERING LIBRARY

SUMMARY OF OBJECTS

```
Object oWebViewFiltersPanel is a cWebViewFiltersPanel
  Set peRegion to prTop

Object oFilterManager is a csWebFilterManager
  Set piFilterPanelLabelOffset to 120
  Set piFilterPanelWidth to 320
End_Object

Object oFilterPanelGroup is a cWebGroup
  Set pbShowCaption to False
  Set psCSSClass to "WebViewCommandBar"

Object oFreeFormFilterPanel is a cWebPanel
  Set peRegion to prLeft
  Set piWidth to 300

Object oSearchForm is a csWebListFilterForm
End_Object
End_Object

Object oFiltersPanel is a cWebPanel
Object oWebCommandBar is a cWebCommandBar
  Set psCSSClass to "WebViewCommandBar Noshadow"

Object oWebToolBar is a cWebToolBar
  Set pbShowCaption to True
  Set pbRender to False

Object oFilterButton is a csWebFilterMenuItem
End_Object

Object oLoadFiltersButton is a csWebLoadFiltersMenuItem
End_Object
End_Object
End_Object
End_Object
End_Object

Object oAppliedFilters is a csWebFilterTagsDisplay
  Set piColumnSpan to 0
End_Object
End_Object
```

SORTING AND FILTERING LIBRARY

SUMMARY OF OBJECTS

```
Object oWebViewFiltersPanel is a cWebViewFiltersPanel
Set peRegion to prTop
```

```
Object oFilterManager is a csWebFilterManager
Set piFilterPanelLabelOffset to 120
Set piFilterPanelWidth to 320
End_Object
```

```
Object oFilterPanelGroup is a cWebGroup
Set pbShowCaption to False
Set psCSSClass to "WebViewCommandBar"
```

```
Object oFreeFormFilterPanel is a cWebPanel
Set peRegion to prLeft
Set piWidth to 300
```

```
Object oSearchForm is a csWebListFilterForm
End_Object
End_Object
```

```
Object oFiltersPanel is a cWebPanel
Object oWebCommandBar is a cWebCommandBar
Set psCSSClass to "WebViewCommandBar Noshadow"
```

```
Object oWebToolBar is a cWebToolBar
Set pbShowCaption to True
Set pbRender to False
```

```
Object oFilterButton is a csWebFilterMenuItem
End_Object
```

```
Object oLoadFiltersButton is a csWebLoadFiltersMenuItem
End_Object
```

```
End_Object
End_Object
End_Object
End_Object
```

```
Object oAppliedFilters is a csWebFilterTagsDisplay
Set piColumnSpan to 0
End_Object
End_Object
```

csWebFilterManager

- piFilterPanelWidth.
- piFilterPanelLabelOffset

SORTING AND FILTERING LIBRARY

SUMMARY OF OBJECTS

```
Object oWebViewFiltersPanel is a cWebViewFiltersPanel
  Set peRegion to prTop

Object oFilterManager is a csWebFilterManager
  Set piFilterPanelLabelOffset to 120
  Set piFilterPanelWidth to 320
End_Object

Object oFilterPanelGroup is a cWebGroup
  Set pbShowCaption to False
  Set psCSSClass to "WebViewCommandBar"

Object oFreeFormFilterPanel is a cWebPanel
  Set peRegion to prLeft
  Set piWidth to 500

Object oSearchForm is a csWebListFilterForm
End_Object
End_Object

Object oFiltersPanel is a cWebPanel
Object oWebCommandBar is a cWebCommandBar
  Set psCSSClass to "WebViewCommandBar Noshadow"

Object oWebToolBar is a cWebToolBar
  Set pbShowCaption to True
  Set pbRender to False

Object oFilterButton is a csWebFilterMenuItem
End_Object

Object oLoadFiltersButton is a csWebLoadFiltersMenuItem
End_Object
End_Object
End_Object
End_Object
End_Object

Object oAppliedFilters is a csWebFilterTagsDisplay
  Set piColumnSpan to 0
End_Object
End_Object
```

csWebListFilterForm

- pbFilterOnHiddenColumns
- piMinSearchTextLength

SORTING AND FILTERING LIBRARY

SUMMARY OF OBJECTS

```
Object oWebViewFiltersPanel is a cWebViewFiltersPanel
  Set peRegion to prTop

Object oFilterManager is a csWebFilterManager
  Set piFilterPanelLabelOffset to 120
  Set piFilterPanelWidth to 320
End_Object

Object oFilterPanelGroup is a cWebGroup
  Set pbShowCaption to False
  Set psCSSClass to "WebViewCommandBar"

Object oFreeFormFilterPanel is a cWebPanel
  Set peRegion to prLeft
  Set piWidth to 300

Object oSearchForm is a csWebListFilterForm
End_Object
End_Object

Object oFiltersPanel is a cWebPanel
Object oWebCommandBar is a cWebCommandBar
  Set psCSSClass to "WebViewCommandBar Noshadow"

Object oWebToolBar is a cWebToolBar
  Set pbShowCaption to True
  Set pbRender to False

Object oFilterButton is a csWebFilterMenuItem
End_Object

Object oLoadFiltersButton is a csWebLoadFiltersMenuItem
End_Object
End_Object
End_Object
End_Object
End_Object

Object oAppliedFilters is a csWebFilterTagsDisplay
  Set piColumnSpan to 0
End_Object
End_Object
```

Filter Buttons

Only visible if filters
are registered

SORTING AND FILTERING LIBRARY

SUMMARY OF OBJECTS

```
Object oWebViewFiltersPanel is a cWebViewFiltersPanel
  Set peRegion to prTop

Object oFilterManager is a csWebFilterManager
  Set piFilterPanelLabelOffset to 120
  Set piFilterPanelWidth to 320
End_Object

Object oFilterPanelGroup is a cWebGroup
  Set pbShowCaption to False
  Set psCSSClass to "WebViewCommandBar"

Object oFreeFormFilterPanel is a cWebPanel
  Set peRegion to prLeft
  Set piWidth to 300

Object oSearchForm is a csWebListFilterForm
End_Object
End_Object

Object oFiltersPanel is a cWebPanel
Object oWebCommandBar is a cWebCommandBar
  Set psCSSClass to "WebViewCommandBar Noshadow"

Object oWebToolBar is a cWebToolBar
  Set pbShowCaption to True
  Set pbRender to False

Object oFilterButton is a csWebFilterMenuItem
End_Object

Object oLoadFiltersButton is a csWebLoadFiltersMenuItem
End_Object
End_Object
End_Object
End_Object
End_Object

Object oAppliedFilters is a csWebFilterTagsDisplay
  Set piColumnSpan to 0
End_Object
End_Object
```

Filter Tags display

/*50 YEARS*/

/*QUESTIONS?*/

