

8.3.1 - Export Boost

Definition

- Boost export speeds export execution from DSCP to DH
- Way of work:
 - 1.Export data in a csv file, using existing export scenario/mapping
 - 2.Load csv file in a temporary table into the DH
 - 3.Insert data from temporary table to destination table
 - 4.Delete csv file

Function BulkGenericExportScenarioInfoById

Structure :

- BulkGenericExportScenarioInfoById(ByRef TextFile, ByRef oDBCON, ByVal nId, ByRef vsFileName, ByRef bRemoveZero, ByRef nPastYearsToKeep, ByRef nFutureMonthsToKeep)

Example :

- BulkGenericExportScenarioInfoById TextFile, oDBCON, 16787388, strBulkFileName, False, 1, 0
- This export is applied to DR export tables

Structure :

- ByRef TextFile: file type
- ByRef oDBCON: DH connection
- ByVal nId: export scenario ID
- ByRef vsFileName: file name
- ByRef bRemoveZero: boolean to manage 0 (case default value <= 0):
 - True : 0 data are deleted into temporary table
 - False: 0 data are kept into temporary table
- ByRef nPastYearsToKeep: past data management (by complete year):
 - 0: data all over the horizon are kept
 - Otherwise formula : delete data where year <= current year- nPastYearsToKeep
 - 1: we delete past data before current year
 - Example current day 24/11/2020 => past data until 2019 included are deleted
 - 2: we delete past data before current year -1
 - Example current day 24/11/2020 => past data until 2018 included are deleted
 - 3 et plus (variable not limited) : 2017 and before
- ByRef nFutureMonthsToKeep : future data management (by month)
 - 0: data all over the horizon are kept
 - Otherwise formula : delete data where month > current month + nFutureMonthsto Keep

Use in a macro :

- Call function strBulkFilename

strBulkFileName = GetUNC(1) & "\" & strSQLConnectionName & \"_\" & GetMacroIdByName(GetCurrentMacroName) & \"_bf\"

- Call export function :

BulkGenericExportScenarioInfoById TextFile, oDBCON, 16787388, strBulkFileName, False, 1

- Export scenario needs to be created before => ID
- Call delete function to delete csv files after export:

DeleteFile(strBulkFileName)

Use in a macro :

- Following functions must be created into « Import – Library – Base Function»
 - GetMacroIdByName
 - DeleteFile
- Path files locating csv files generated are defined into « GetUNC»

- strAppInterfacesCusPI / TI / SI / DI
- Corresponding to GetUNC(1)

Datahub :

- Use of « sys_bulk» temporary tables

```
"truncate table sys_bulk_" & nDim & "d_" & strSQLConnectionName
```

```

+ dbo.Sys_Bulk_2d_dynasys_dip1
+ dbo.Sys_Bulk_2d_dynasys_dip2
+ dbo.Sys_Bulk_2d_dynasys_dip3
+ dbo.Sys_Bulk_2d_dynasys_dp1
+ dbo.Sys_Bulk_2d_dynasys_dp2
+ dbo.Sys_Bulk_2d_dynasys_dp3
+ dbo.Sys_Bulk_3d_dynasys_dip1
+ dbo.Sys_Bulk_3d_dynasys_dip2
+ dbo.Sys_Bulk_3d_dynasys_dip3
+ dbo.Sys_Bulk_3d_dynasys_dp1
+ dbo.Sys_Bulk_3d_dynasys_dp2
+ dbo.Sys_Bulk_3d_dynasys_dp3

```

Execution details :

- Step 1: data export (except default values) in a csv file with name of scenario on the server => data are exported following datafields dimensions order

```
SaveParamAsCSVFileWithoutDefaults nParamId, ";", vsFileName, True
```

- Step 2: function determines the datafield number of dimensions and delete corresponding temporary table

```
"truncate table sys_bulk_" & nDim & "d_" & strSQLConnectionName
```

- Step 3: csv file integration into corresponding temporary table

```
"bulk insert sys_bulk_" & nDim & "d_" & strSQLConnectionName & " from '" & vsFileName & "' with (fieldtermina
```

- Step 4: management of 0 values into temporary table

```

If bRemoveZero Then
  ExecuteQuery TextFile, oDBCConn, "delete from sys_bulk_" & nDim & "d_" & strSQLConnectionName & " where Field" & nDim+1 & " ='0'"
End If

```

Execution details :

- Step 5: past data management

```

If nPastYearsToKeep > 0 Then
  ExecuteQuery TextFile, oDBCConn, "delete from sys_bulk_" & nDim & "d_" & strSQLConnectionName & " where right(Field" & nDim & ",4) <= year(getdate())-" & nPastYearsToKeep
End If

```

- Step 6 : future data management

```

If nFutureMonthsToKeep > 0 Then
  ExecuteQuery TextFile, oDBCConn, "delete from sys_bulk_" & nDim & "d_" & strSQLConnectionName & " where convert(date, '01/'+right(left (Field" & nDim & ", 3), 2)+'/'+right(Field" & nDim & ",4), 103) > dateadd(month," & nFutureMonthsToKeep
End If

```

- Step 7: delete destination table

- The function uses the name of ODBC query to call corresponding SQL view => they must have the same name to feed corresponding destination DH table

```
"delete from " & oScenario.Query
```

- Step 8: insert data from temporary to destination DH table
 - The function uses also the name of the ODBC query to call SQL view

```
"insert into " & oScenario.Query & " select '" & strSQLConnectionName & "', '', '', * from sys_bulk_" & nDim & "d_" & strSQLConnectionName
```

- In the case above, function inserts :
 - strSQLConnectionName
 - 2 empty fields
 - Then fields of temporary table

Execution details :

- Step 7:
 - Temporary and destination tables structures are different => insert structure is specific
 - Insert doesn't manage the order of data between temporary and destination tables => data order when generating csv file must fit with destination table fields order
 - In DR export case, temporary tables and DR tables consist of data types => VARCHAR => no problem within insert
- Global :
 - If a selection of data needs to be adjusted, function variables need to be modified and not the mapping scenario (the best is to align both)

Deployment – Pre-requisite

- Can be applied to export with execution during more than 10s
- To be checked before:

1.Scenario export settings :

- 1.Selection on horizon, sku...
- 2.Default values + null values settings
- 3.Destination table reset

2.Exported datafield dimensions order vs destination table

3.Date format of horizon period

4.SQL table or view and corresponding ODBC query names – similar names without space and parenthesis

5.Temporary tables corresponding to number of dimensions to be exported

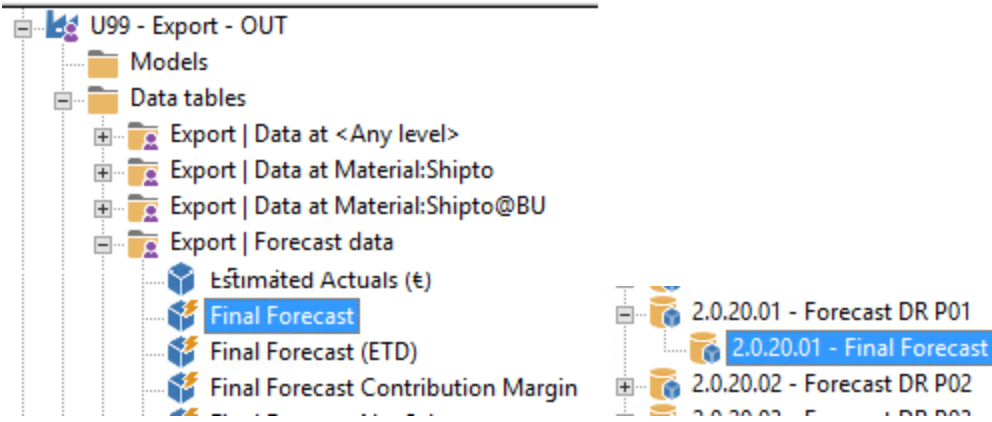
6.Data type consistent between temporary and destination tables

7.Insert structure corresponding to destination tables

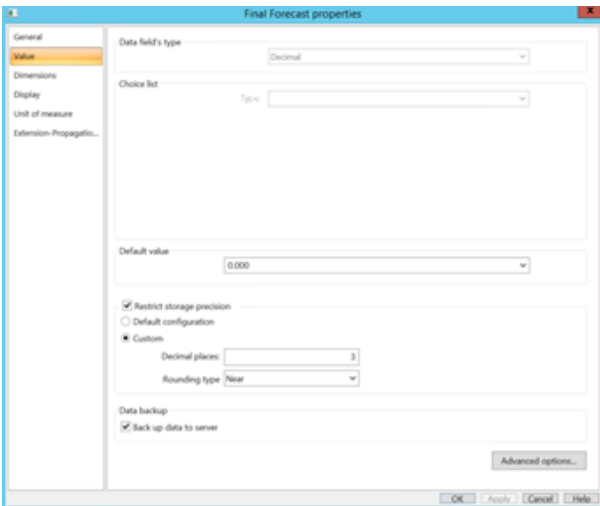
- The function is currently adapted to QV and DR SQL tables
- The code will probably have to be updated in case of deployment of this function on other exports

Step by step

- Step 1 : identify an export to be boosted
 - Example: execution > 10s (info in log file)



- Step 2 : check pre-requisite
 - Datafield settings:
 - Default value = 0



- Step 2 : identify settings of this export
 - Datafield settings:
 - Dimensions order conform to destination export table : DFU_Code (Item) first then Period (Horizon)

Final Forecast properties

Type	Name	Dimension	Condition (1st)	Condition (2nd)	Active	Aggregation
Item	Material:Shipto	Material:Shipto@DC	Active DFUs		<input checked="" type="checkbox"/>	Sum
Horizon	Horizon	Forecast month	Calendar days		<input checked="" type="checkbox"/>	Sum

Results | Messages

App_Name	Table_Source	Table_Destination	DFU_Code	Period	Quantity
dynasys_dp3			121431R:64218R@06R	M11-2021	25200
dynasys_dp3			121431R:64218R@06R	M01-2022	25200
dynasys_dp3			121431R:64218R@06R	M02-2022	50400

- Step 2 : identify settings of this export

- Current mapping settings:
 - Default values not exported only
 - Past horizon before current year not exported

2.0.20.01 - Final Forecast properties

- Final Forecast
- Final Forecast (ETD)
- Final Forecast Contribution Mar
- Final Forecast Net Sales
- Final Forecast Net Sales [LC]
- Final Forecast NetBack Sales
- Final Sales Price

Data field

Don't export default values

Don't export zero values

Selection
Selection

Choose when executing

Dimension : Horizon (*)

M01-2017	M01-2020
M02-2017	M02-2020
M03-2017	M03-2020
M04-2017	M04-2020
M05-2017	M05-2020
M06-2017	M06-2020
M07-2017	M07-2020
M08-2017	M08-2020
M09-2017	M09-2020
M10-2017	M10-2020
M11-2017	M11-2020
M12-2017	M12-2020
M01-2018	M01-2021
M02-2018	M02-2021
M03-2018	M03-2021
M04-2018	M04-2021
M05-2018	M05-2021
M06-2018	M06-2021
M07-2018	M07-2021
M08-2018	M08-2021
M09-2018	M09-2021
M10-2018	M10-2021
M11-2018	M11-2021
M12-2018	M12-2021
M01-2019	M01-2022
M02-2019	M02-2022
M03-2019	M03-2022
M04-2019	M04-2022
M05-2019	M05-2022
M06-2019	M06-2022
M07-2019	M07-2022
M08-2019	M08-2022
M09-2019	M09-2022
M10-2019	M10-2022
M11-2019	M11-2022
M12-2019	M12-2022

- Table is reset before export + SQL view and ODBC query have the same name :

Macros: TECH : Export Forecast Demand Review

```

For i = 1 To 76
  ExecuteQuery TextFile, oDBCEN, "delete from EXP_FORE_DR_P" & right("0" & i, 2)
Next

```

- Step 3 : modify export macro
 - Set up macro with function boost:

```

Macros: TECH : Export Forecast Demand Review TEST

Option Explicit

Dim TextFile

' ExecuteMacro "Import - Library"
ExecuteMacroById 16787236

Set TextFile = New LogFile
TextFile.CreateLog GetCurrentMacroName
MainUpdate TextFile, True
Set TextFile = Nothing

Sub MainUpdate(ByRef TextFile, ByVal bQuiet)
    Dim oDBCON
    Dim nRecord
    Dim strPath, HasErrors, strBulkFileName
    Dim i

    DBConnectImportExport TextFile, oDBCON

    DataHubLogStart TextFile, oDBCON, True

    strBulkFileName = GetUNC(1) & "\" & strSQLConnectionName & "_" & GetMacroIdByName(GetCurrentMacroName) & ".csv"
    '2.0.20.01 - Final Forecast (E01)
    BulkGenericExportScenarioInfoById TextFile, oDBCON, 16787388, strBulkFileName, False, 1

    DeleteFile(strBulkFileName)

    DataHubLogEnd TextFile, oDBCON, False, True

    If Not TextFile Is Nothing Then TextFile.SetTraceOff oDBCON
    DBDisconnect oDBCON
End Sub

```

- Declare variable
- Set up file name function
- Replace current scenario mapping execution by new bulk mapping scenario:
 - 0 management : initial export scenario allows 0 export => variable = False
 - Past data : initial export filters past horizon => variable = 1 to delete past horizon before current year
 - Function will reset destination table before insert
- Add function to delete csv file after execution

Definition

- Boost export speeds export execution from DSCP to DH
- Way of work:
 - 1.Export data in a csv file, using existing export scenario/mapping
 - 2.Load csv file in a temporary table into the DH
 - 3.Insert data from temporary table to destination table
 - 4.Delete csv file

Function BulkGenericExportScenarioInfoById

Structure :

- BulkGenericExportScenarioInfoById(ByRef TextFile, ByRef oDBCON, ByVal nId, ByRef vsFileName, ByRef bRemoveZero, ByRef nPastYearsToKeep, ByRef nFutureMonthsToKeep)

Example :

- BulkGenericExportScenarioInfoById TextFile, oDBCON, 16787388, strBulkFileName, False, 1, 0
- This export is applied to DR export tables

Structure :

- ByRef TextFile: file type
- ByRef oDBCON: DH connection
- ByVal nId: export scenario ID
- ByRef vsFileName: file name
- ByRef bRemoveZero: boolean to manage 0 (case default value <> 0):
 - True : 0 data are deleted into temporary table
 - False: 0 data are kept into temporary table
- ByRef nPastYearsToKeep: past data management (by complete year):
 - 0: data all over the horizon are kept
 - Otherwise formula : delete data where year <= current year- nPastYearsToKeep
 - 1: we delete past data before current year
 - Example current day 24/11/2020 => past data until 2019 included are deleted
 - 2: we delete past data before current year -1

- Example current day 24/11/2020 => past data until 2018 included are deleted
 - 3 et plus (variable not limited) : 2017 and before
 - ByRef nFutureMonthsToKeep : future data management (by month)
 - 0: data all over the horizon are kept
 - Otherwise formula : delete data where month > current month + nFutureMonthsto Keep

Use in a macro :

- Call function strBulkFilename

```
strBulkFileName = GetUNC(1) & "\" & strSQLConnectionName & "_" & GetMacroIdByName(GetCurrentMacroName) & "_bf"
```

- Call export function :

```
BulkGenericExportScenarioInfoById TextFile, oDBCON, 16787388, strBulkFileName, False, 1
```

- Export scenario needs to be created before => ID
- Call delete function to delete csv files after export:

```
DeleteFile(strBulkFileName)
```

Use in a macro :

- Following functions must be created into « Import – Library – Base Function»
 - GetMacroIdByName
 - DeleteFile
- Path files locating csv files generated are defined into « GetUNC»
 - strAppInterfacesCusPI / TI / SI / DI
 - Corresponding to GetUNC(1)

Datahub :

- Use of « sys_bulk» temporary tables

```
"truncate table sys_bulk_" & nDim & "d_" & strSQLConnectionName
```

+	dbo.Sys_Bulk_2d_dynasys_dip1
+	dbo.Sys_Bulk_2d_dynasys_dip2
+	dbo.Sys_Bulk_2d_dynasys_dip3
+	dbo.Sys_Bulk_2d_dynasys_dp1
+	dbo.Sys_Bulk_2d_dynasys_dp2
+	dbo.Sys_Bulk_2d_dynasys_dp3
+	dbo.Sys_Bulk_3d_dynasys_dip1
+	dbo.Sys_Bulk_3d_dynasys_dip2
+	dbo.Sys_Bulk_3d_dynasys_dip3
+	dbo.Sys_Bulk_3d_dynasys_dp1
+	dbo.Sys_Bulk_3d_dynasys_dp2
+	dbo.Sys_Bulk_3d_dynasys_dp3

Execution details :

- Step 1: data export (except default values) in a csv file with name of scenario on the server => data are exported following datafields dimensions order

```
SaveParamAsCSVFileWithoutDefaults nParamId, ";", vsFileName, True
```

- Step 2: function determines the datafield number of dimensions and delete corresponding temporary table

```
"truncate table sys_bulk_" & nDim & "d_" & strSQLConnectionName
```

- Step 3: csv file integration into corresponding temporary table

```
"bulk insert sys_bulk_" & nDim & "d_" & strSQLConnectionName & " from '" & vsFileName & "' with (fieldtermina
```

- Step 4: management of 0 values into temporary table

```
-----
If bRemoveZero Then
  ExecuteQuery TextFile, oDBCConn, "delete from sys_bulk_" & nDim & "d_" & strSQLConnectionName & " where Field" & nDim+1 & " ='0'"
End If
```

Execution details :

- Step 5: past data management

```
If nPastYearsToKeep > 0 Then
  ExecuteQuery TextFile, oDBCConn, "delete from sys_bulk_" & nDim & "d_" & strSQLConnectionName & " where right(Field" & nDim & ",4) <= year(getdate())-" & nPastYearsToKeep
End If
```

- Step 6: future data management

```
If nFutureMonthsToKeep > 0 Then
  ExecuteQuery TextFile, oDBCConn, "delete from sys_bulk_" & nDim & "d_" & strSQLConnectionName & " where convert(date, '01/'+right(left (Field" & nDim & ", 3), 2)+'/'+right(Field" & nDim & ",4), 103) > dateadd(month," & nFutureMonthsToKeep
End If
```

- Step 7: delete destination table
 - The function uses the name of ODBC query to call corresponding SQL view => they must have the same name to feed corresponding destination DH table

```
"delete from " & oScenario.Query
```

- Step 8: insert data from temporary to destination DH table
 - The function uses also the name of the ODBC query to call SQL view

```
"insert into " & oScenario.Query & " select '" & strSQLConnectionName & "', '', '', * from sys_bulk_" & nDim & "d_" & strSQLConnectionName
```

- In the case above, function inserts :
 - strSQLConnectionName
 - 2 empty fields
 - Then fields of temporary table

Execution details :

- Step 7:
 - Temporary and destination tables structures are different => insert structure is specific
 - Insert doesn't manage the order of data between temporary and destination tables => data order when generating csv file must fit with destination table fields order
 - In DR export case, temporary tables and DR tables consistents data types => VARCHAR => no problem within insert
- Global :
 - If a selection of data needs to be adjusted, function variables need to be modified and not the mapping scenario (the best is to align both)

Deployment – Pre-requisite

- Can be applied to export with execution during more than 10s
- To be checked before:

1.Scenario export settings :

- 1.Selection on horizon, sku...
- 2.Default values + null values settings
- 3.Destination table reset

2.Exported datafield dimensions order vs destination table

3.Date format of horizon period

4.SQL table or view and corresponding ODBC query names – similar names without space and parenthesis

5.Temporary tables corresponding to number of dimensions to be exported

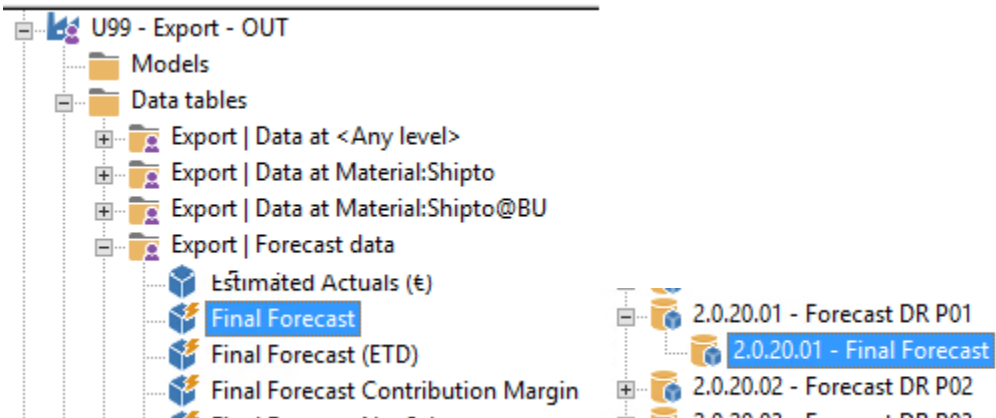
6.Data type consistent between temporary and destination tables

7.Insert structure corresponding to destination tables

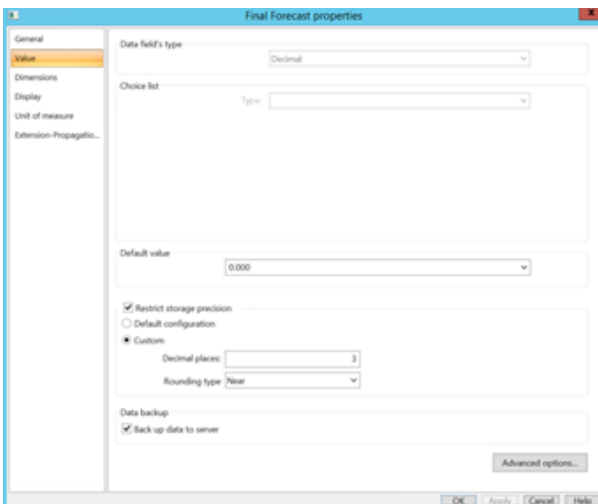
- The function is currently adapted to QV and DR SQL tables
- The code will probably have to updated in case of deployment of this function on other exports

Step by step

- Step 1 : identify an export to be boosted
 - Example: execution > 10s (info in log file)



- Step 2 : check pre-requisite
 - Datafield settings:
 - Default value = 0



- Step 2 : identify settings of this export
 - Datafield settings:
 - Dimensions order conform to destination export table : DFU_Code (Item) first then Period (Horizon)

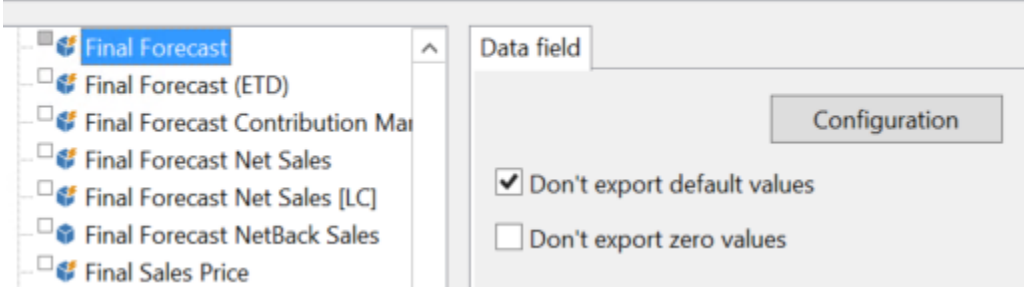
Final Forecast properties

Type	Name	Dimension	Condition (1st)	Condition (2nd)	Active	Agg
Item	Material:Shipto	Material:Shipto@DC	Active DFUs		<input checked="" type="checkbox"/>	Sum
Horizon	Horizon	Forecast month	Calendar days		<input checked="" type="checkbox"/>	Sum

Results		Messages			
App_Name	Table_Source	Table_Destination	DFU_Code	Period	Quantity
dynasys_dp3			121431R:64218R@06R	M11-2021	25200
dynasys_dp3			121431R:64218R@06R	M01-2022	25200
dynasys_dp3			121431R:64218R@06R	M02-2022	50400

- Step 2 : identify settings of this export
 - Current mapping settings:
 - Default values not exported only
 - Past horizon before current year not exported

2.0.20.01 - Final Forecast properties



Final Forecast

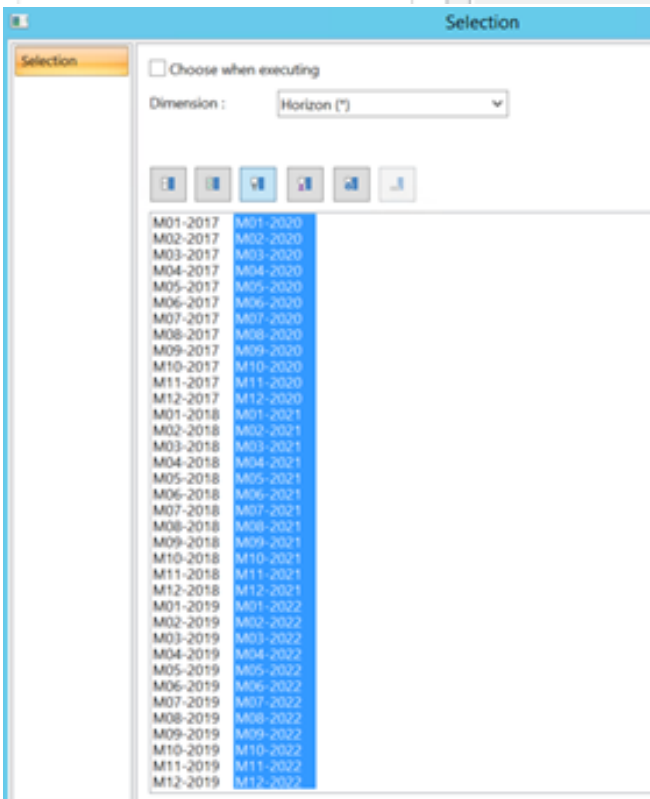
- Final Forecast (ETD)
- Final Forecast Contribution Mar
- Final Forecast Net Sales
- Final Forecast Net Sales [LC]
- Final Forecast NetBack Sales
- Final Sales Price

Data field

Configuration

Don't export default values

Don't export zero values



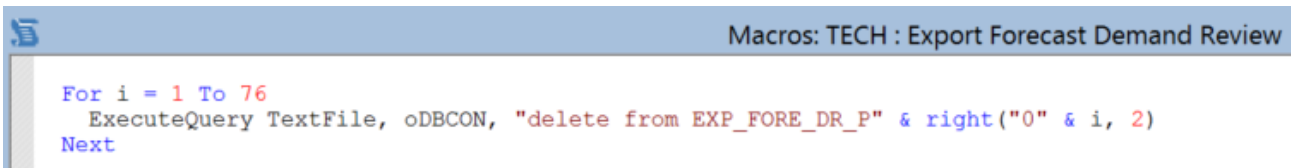
Selection

Choose when executing

Dimension : Horizon (*)

M01-2017	M01-2020
M02-2017	M02-2020
M03-2017	M03-2020
M04-2017	M04-2020
M05-2017	M05-2020
M06-2017	M06-2020
M07-2017	M07-2020
M08-2017	M08-2020
M09-2017	M09-2020
M10-2017	M10-2020
M11-2017	M11-2020
M12-2017	M12-2020
M01-2018	M01-2021
M02-2018	M02-2021
M03-2018	M03-2021
M04-2018	M04-2021
M05-2018	M05-2021
M06-2018	M06-2021
M07-2018	M07-2021
M08-2018	M08-2021
M09-2018	M09-2021
M10-2018	M10-2021
M11-2018	M11-2021
M12-2018	M12-2021
M01-2019	M01-2022
M02-2019	M02-2022
M03-2019	M03-2022
M04-2019	M04-2022
M05-2019	M05-2022
M06-2019	M06-2022
M07-2019	M07-2022
M08-2019	M08-2022
M09-2019	M09-2022
M10-2019	M10-2022
M11-2019	M11-2022
M12-2019	M12-2022

- Table is reset before export + SQL view and ODBC query have the same name :



Macros: TECH : Export Forecast Demand Review

```
For i = 1 To 76
    ExecuteQuery TextFile, oDBCConn, "delete from EXP_FORE_DR_P" & right("0" & i, 2)
Next
```

- Step 3 : modify export macro
 - Set up macro with function boost:

```

Option Explicit

Dim TextFile

' ExecuteMacro "Import - Library"
ExecuteMacroById 16787236

Set TextFile = New LogFile
TextFile.CreateLog GetCurrentMacroName
MainUpdate TextFile, True
Set TextFile = Nothing

Sub MainUpdate(ByRef TextFile, ByVal bQuiet)
    Dim oDBCConn
    Dim nRecord
    Dim strPath, HasErrors, strBulkFileName
    Dim i

    DBConnectImportExport TextFile, oDBCConn

    DataHubLogStart TextFile, oDBCConn, True

    strBulkFileName = GetUNC(1) & "\\" & strSQLConnectionName & "_" & GetMacroIdByName(GetCurrentMacroName) & ".csv"

    '2.0.20.01 - Final Forecast (E01)
    BulkGenericExportScenarioInfoById TextFile, oDBCConn, 16787388, strBulkFileName, False, 1

    DeleteFile(strBulkFileName)

    DataHubLogEnd TextFile, oDBCConn, False, True

    If Not TextFile Is Nothing Then TextFile.SetTraceOff oDBCConn
    DBDisconnect oDBCConn
End Sub

```

- Declare variable
- Set up file name function
- Replace current scenario mapping execution by new bulk mapping scenario:
 - 0 management : initial export scenario allows 0 export => variable = False
 - Past data : initial export filters past horizon => variable = 1 to delete past horizon before current year
 - Function will reset destination table before insert
- Add function to delete csv file after execution