BI4Dynamics Customization Manual

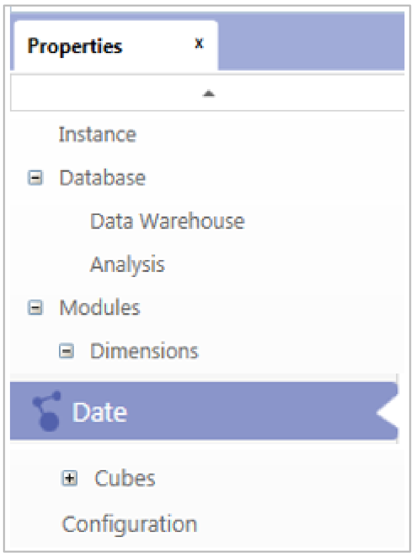
Last update: August 2017
BI4Dynamics NAV version 5.6.0
Revision 2

1 SETTING UP INSTANCE

Upon opening the instance, proceed to Instance properties by clicking File and Properties.

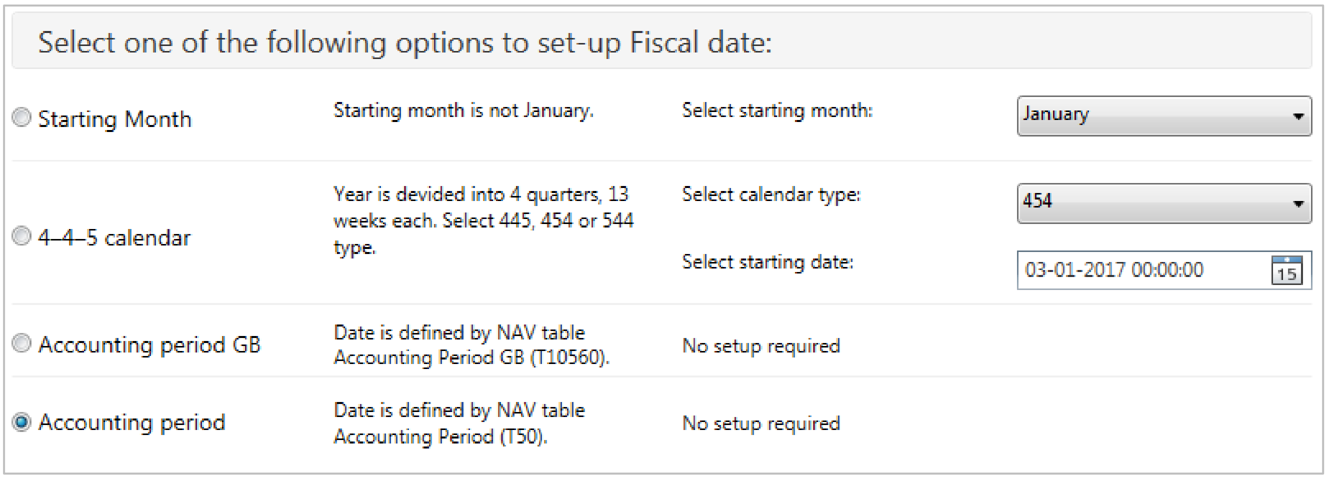
1.1 Setup dimensions

1.1.1 Set Date



New BI4Dynamics feature (Fiscal Year Date Dimension) allows you to analyze your data by Fiscal Date instead of the calendar date.
Set the Fiscal Date Offset in line with the start of your Fiscal Year or choose the 4-4-5 calendar option, which enables you to analyses your data by selected quarters.

- Example 1: If your Fiscal Year starts with April, set the Fiscal Date Offset to 3.
- Example 2: If your Fiscal Year starts with July, set the Fiscal Date Offset to 6.
- Example 3: Select one of the options (4-4-5, 4-5-4 or 5-4-4) and the starting date

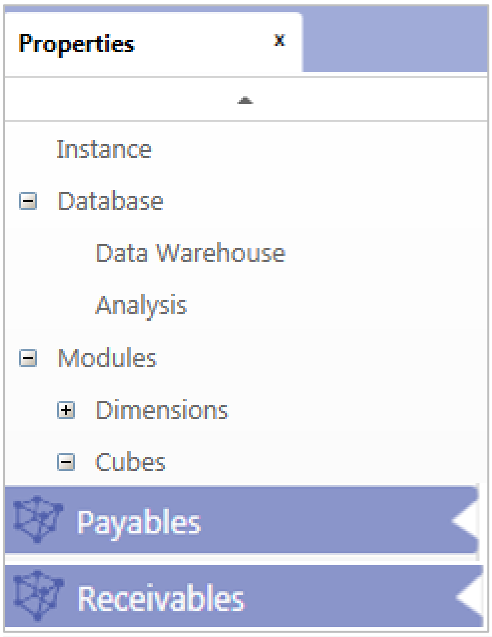


Select one of the following options to set-up Fiscal date:

○ Starting Month	Starting month is not January.	Select starting month:	January
○ 4–4–5 calendar	Year is devided into 4 quarters, 13 weeks each. Select 445, 454 or 544 type.	Select calendar type: Select starting date:	454 03-01-2017 00:00:00
○ Accounting period GB	Date is defined by NAV table Accounting Period GB (T10560).	No setup required	
● Accounting period	Date is defined by NAV table Accounting Period (T50).	No setup required	

1.2 Setup cubes

1.2.1 Set Receivables or Payables compression

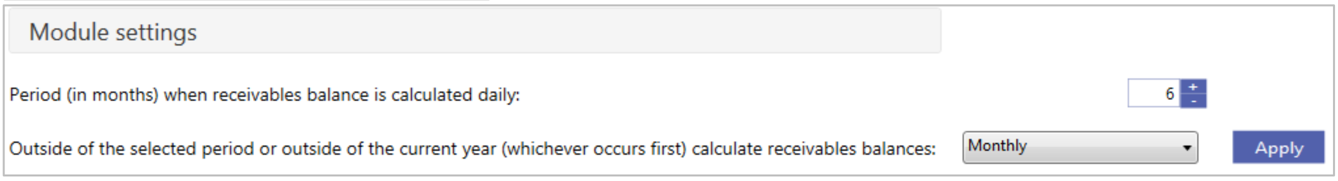


New BI4Dynamics feature (set Receivables or/and Payables compression) allows to set the number of months in which the receivables or payables balance is calculated on a daily basis.

Defining how the balance is calculated before the selected period is limited to Monthly or Yearly option, but Daily option can be used as well to keep the calculation on a daily level.

Limiting the receivables or payables balance calculation will greatly improve the processing times.

The default setting for this new feature can be seen bellow.



Module settings

Period (in months) when receivables balance is calculated daily: 6

Outside of the selected period or outside of the current year (whichever occurs first) calculate receivables balances: Monthly

1.3 Manage stage

1.3.1 Filter data stage

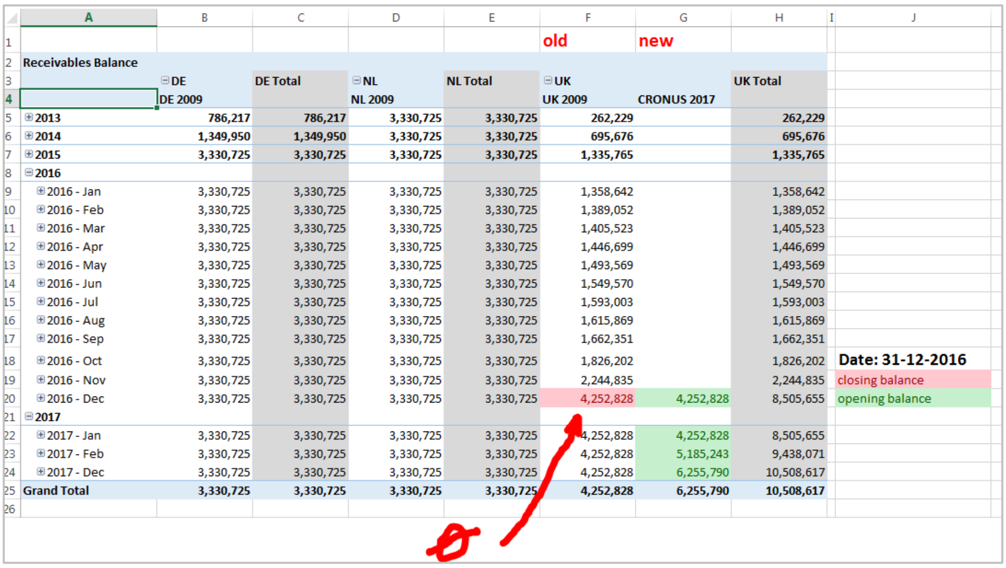
A brand new BI4Dynamics functionality is introduced in version 5.4. You now have the ability to filter the transaction data loaded from one or more NAV databases or companies to BI4Dynamics stage.

To better understand this feature we prepared a simple example its usability. We will show you how to filter closing receivables balance in archive database without altering data in NAV:

1 CHECK BALANCES

1. Check that ending balance in old company equals to opening balance in new company
    - Use Dynamics NAV report and Excel from BI4Dynamics cube
    - If balances do not match – stop!
    - Get it sorted out

Example:



A	B	C	D	E	F	G	H	I	J
					old	new			
Receivables Balance									
	⊟DE	DE Total	⊟NL	NL Total	⊟UK		UK Total		
	DE 2009		NL 2009		UK 2009	CRONUS 2017			
⊞2013	786,217	786,217	3,330,725	3,330,725	262,229		262,229		
⊞2014	1,349,950	1,349,950	3,330,725	3,330,725	695,676		695,676		
⊞2015	3,330,725	3,330,725	3,330,725	3,330,725	1,335,765		1,335,765		
⊟2016									
⊞2016 - Jan	3,330,725	3,330,725	3,330,725	3,330,725	1,358,642		1,358,642		
⊞2016 - Feb	3,330,725	3,330,725	3,330,725	3,330,725	1,389,052		1,389,052		
⊞2016 - Mar	3,330,725	3,330,725	3,330,725	3,330,725	1,405,523		1,405,523		
⊞2016 - Apr	3,330,725	3,330,725	3,330,725	3,330,725	1,446,699		1,446,699		
⊞2016 - May	3,330,725	3,330,725	3,330,725	3,330,725	1,493,569		1,493,569		
⊞2016 - Jun	3,330,725	3,330,725	3,330,725	3,330,725	1,549,570		1,549,570		
⊞2016 - Jul	3,330,725	3,330,725	3,330,725	3,330,725	1,593,003		1,593,003		
⊞2016 - Aug	3,330,725	3,330,725	3,330,725	3,330,725	1,615,869		1,615,869		
⊞2016 - Sep	3,330,725	3,330,725	3,330,725	3,330,725	1,662,351		1,662,351		
⊞2016 - Oct	3,330,725	3,330,725	3,330,725	3,330,725	1,826,202		1,826,202		Date: 31-12-2016
⊞2016 - Nov	3,330,725	3,330,725	3,330,725	3,330,725	2,244,835		2,244,835		closing balance
⊞2016 - Dec	3,330,725	3,330,725	3,330,725	3,330,725	4,252,828	4,252,828	8,505,655		opening balance
⊟2017									
⊞2017 - Jan	3,330,725	3,330,725	3,330,725	3,330,725	4,252,828	4,252,828	8,505,655		
⊞2017 - Feb	3,330,725	3,330,725	3,330,725	3,330,725	4,252,828	5,185,243	9,438,071		
⊞2017 - Dec	3,330,725	3,330,725	3,330,725	3,330,725	4,252,828	6,255,790	10,508,617		
Grand Total	3,330,725	3,330,725	3,330,725	3,330,725	4,252,828	6,255,790	10,508,617		

Status*:*

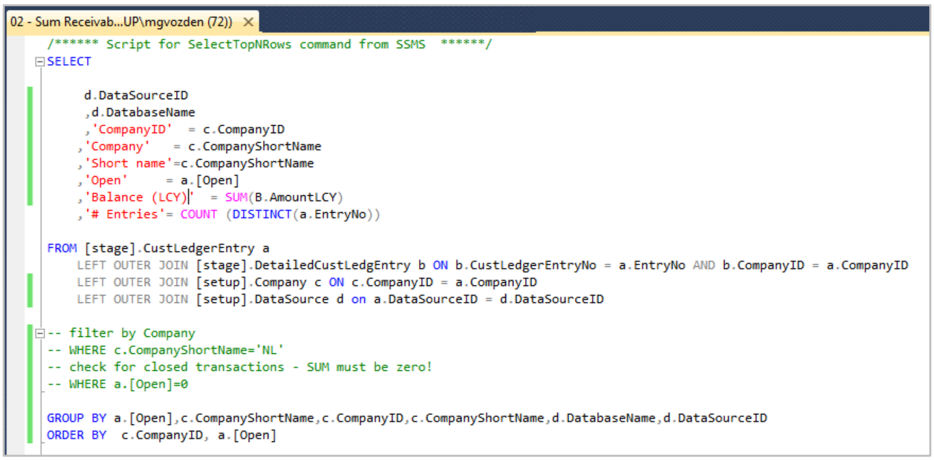
Closing balance in old company - UK 2009 equals Opening balance in Cronus 2017. This is OK

Goal of this project:

Closing balance in old company - UK2009 - should be set to zero in BI and keep data in NAV unchanged.

2. Check Receivables Balance in SQL (run SQL report)
    - Total Balance must match NAV (per company)
    - Balance of all open entries must match total Balance
    - Balance of all closed entries must be zero
    - If Balances do not match – stop!

- Get it sorted out



```sql
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT

     d.DataSourceID
     ,d.DatabaseName
     ,'CompanyID'  = c.CompanyID
    ,'Company'   = c.CompanyShortName
    ,'Short name'=c.CompanyShortName
    ,'Open'      = a.[Open]
    ,'Balance (LCY)'  = SUM(B.AmountLCY)
    ,'# Entries'= COUNT (DISTINCT(a.EntryNo))

FROM [stage].CustLedgerEntry a
    LEFT OUTER JOIN [stage].DetailedCustLedgEntry b ON b.CustLedgerEntryNo = a.EntryNo AND b.CompanyID = a.CompanyID
    LEFT OUTER JOIN [setup].Company c ON c.CompanyID = a.CompanyID
    LEFT OUTER JOIN [setup].DataSource d on a.DataSourceID = d.DataSourceID

-- filter by Company
-- WHERE c.CompanyShortName='NL'
-- check for closed transactions - SUM must be zero!
-- WHERE a.[Open]=0

GROUP BY a.[Open],c.CompanyShortName,c.CompanyID,c.CompanyShortName,d.DatabaseName,d.DataSourceID
ORDER BY  c.CompanyID, a.[Open]
```

Example

2 FILTER STAGE

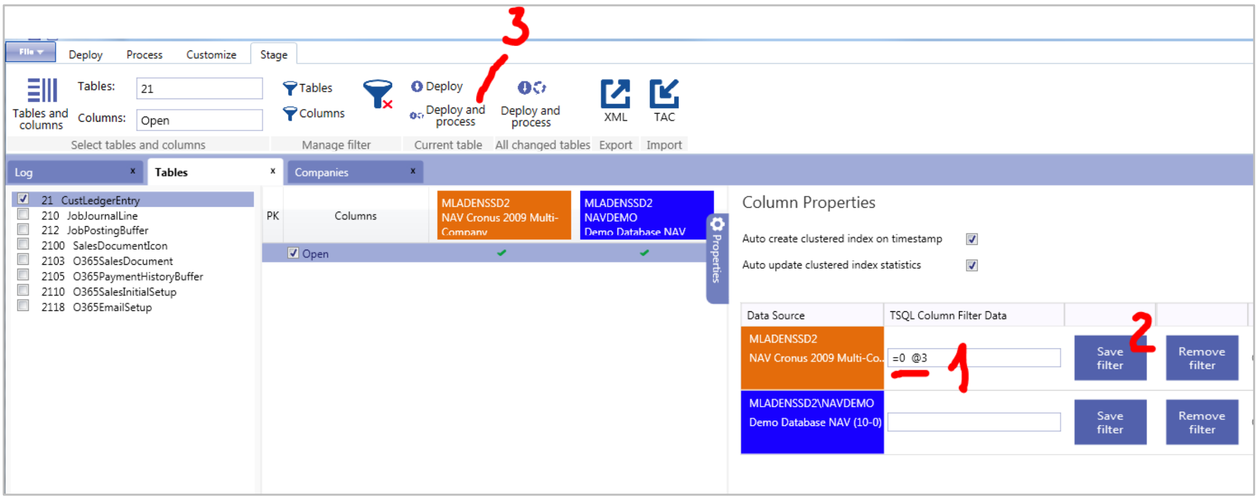
3. Go to filter area and select old data source (your selection is on T=21/F=Open)
4. Set value: **=0** (*see picture #1)*

   - only closed entries will be copied from NAV
   - As we filter integer type of field, so there is no '0' or "0", only 0.
   - Set the number of the company that you apply this rule
   - =0 @3 (this will apply filter for CompanyID=3, an get all data from other companies with filter into BI4Dynamics)

5. Run "Save filter" (see *picture #2)*
6. Run "Deploy & Process" on Current Table (see *picture #3*)

   - Deploy will apply filter to the stage
   - Process will select data from NAV

Example:



Tables: 21

Columns: Open

☑ 21 CustLedgerEntry
☐ 210 JobJournalLine
☐ 212 JobPostingBuffer
☐ 2100 SalesDocumentIcon
☐ 2103 O365SalesDocument
☐ 2105 O365PaymentHistoryBuffer
☐ 2110 O365SalesInitialSetup
☐ 2118 O365EmailSetup

PK	Columns	MLADENSSD2 NAV Cronus 2009 Multi-Company	MLADENSSD2 NAVDEMO Demo Database NAV
	☑ Open	✓	✓

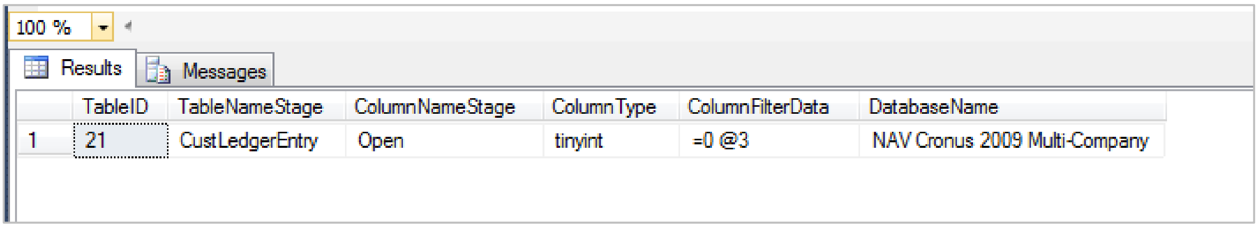
Auto create clustered index on timestamp ☑

Auto update clustered index statistics ☑

Data Source	TSQL Column Filter Data
MLADENSSD2 NAV Cronus 2009 Multi-Co..	=0 @3
MLADENSSD2\NAVDEMO Demo Database NAV (10-0)	

7. Check filters on staging area – this step is optional (**run SQL report)**

Example:



	TableID	TableNameStage	ColumnNameStage	ColumnType	ColumnFilterData	DatabaseName
1	21	CustLedgerEntry	Open	tinyint	=0 @3	NAV Cronus 2009 Multi-Company

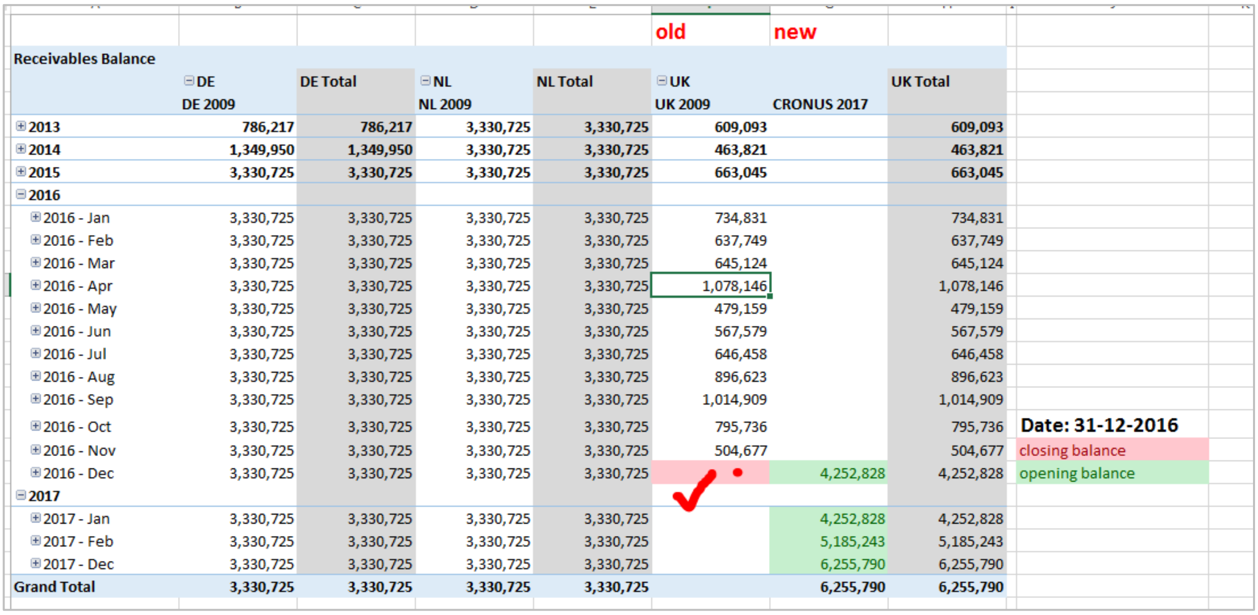
8. Process DW, cubes

3 CONTROL

9. Check report in Excel

- Receivables balance in old company (the one that we have filtered) should be zero!

Example:



					old	new		
Receivables Balance								
	DE	DE Total	NL	NL Total	UK		UK Total	
	DE 2009		NL 2009		UK 2009	CRONUS 2017		
2013	786,217	786,217	3,330,725	3,330,725	609,093		609,093	
2014	1,349,950	1,349,950	3,330,725	3,330,725	463,821		463,821	
2015	3,330,725	3,330,725	3,330,725	3,330,725	663,045		663,045	
2016								
2016 - Jan	3,330,725	3,330,725	3,330,725	3,330,725	734,831		734,831	
2016 - Feb	3,330,725	3,330,725	3,330,725	3,330,725	637,749		637,749	
2016 - Mar	3,330,725	3,330,725	3,330,725	3,330,725	645,124		645,124	
2016 - Apr	3,330,725	3,330,725	3,330,725	3,330,725	1,078,146		1,078,146	
2016 - May	3,330,725	3,330,725	3,330,725	3,330,725	479,159		479,159	
2016 - Jun	3,330,725	3,330,725	3,330,725	3,330,725	567,579		567,579	
2016 - Jul	3,330,725	3,330,725	3,330,725	3,330,725	646,458		646,458	
2016 - Aug	3,330,725	3,330,725	3,330,725	3,330,725	896,623		896,623	
2016 - Sep	3,330,725	3,330,725	3,330,725	3,330,725	1,014,909		1,014,909	
2016 - Oct	3,330,725	3,330,725	3,330,725	3,330,725	795,736		795,736	Date: 31-12-2016
2016 - Nov	3,330,725	3,330,725	3,330,725	3,330,725	504,677		504,677	closing balance
2016 - Dec	3,330,725	3,330,725	3,330,725	3,330,725	✓	4,252,828	4,252,828	opening balance
2017								
2017 - Jan	3,330,725	3,330,725	3,330,725	3,330,725		4,252,828	4,252,828	
2017 - Feb	3,330,725	3,330,725	3,330,725	3,330,725		5,185,243	5,185,243	
2017 - Dec	3,330,725	3,330,725	3,330,725	3,330,725		6,255,790	6,255,790	
Grand Total	3,330,725	3,330,725	3,330,725	3,330,725		6,255,790	6,255,790	

4 APPENDIX

If you need to filter more companies from same data source that add CompanyID by comma:
=0 @3,4
This will apply filter for CompanyID=3 and 4

Filter has to be set per data source.
=0 @1,3,4

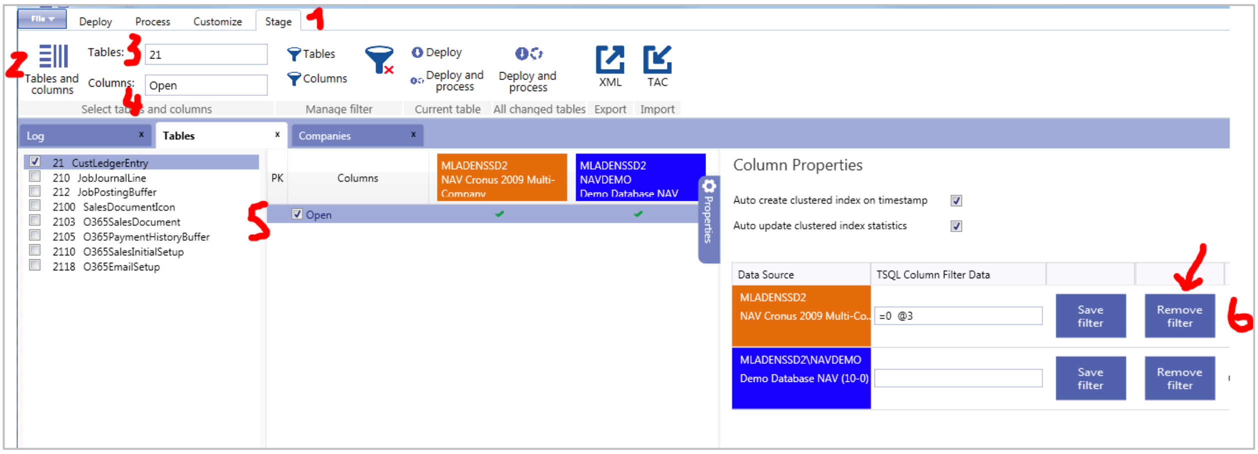
If CompanyID=1 is in other data source than filter will not apply to CompanyID=1.

5 TROUBLESHOOTING

If something went wrong, you can delete filter at any time without effecting NAV.

10. Go to Stage\\Tables & Columns\\Tables=21, Columns=Open\\ select the column Open
11. Run Remove filter (*see picture #6*)

Example:



Tables: 21

Columns: Open

☑ 21 CustLedgerEntry
□ 210 JobJournalLine
□ 212 JobPostingBuffer
□ 2100 SalesDocumentIcon
□ 2103 O365SalesDocument
□ 2105 O365PaymentHistoryBuffer
□ 2110 O365SalesInitialSetup
□ 2118 O365EmailSetup

PK	Columns	MLADENSSD2 NAV Cronus 2009 Multi-Company	MLADENSSD2 NAVDEMO Demo Database NAV
	☑ Open	✓	✓

Column Properties

Auto create clustered index on timestamp ☑

Auto update clustered index statistics ☑

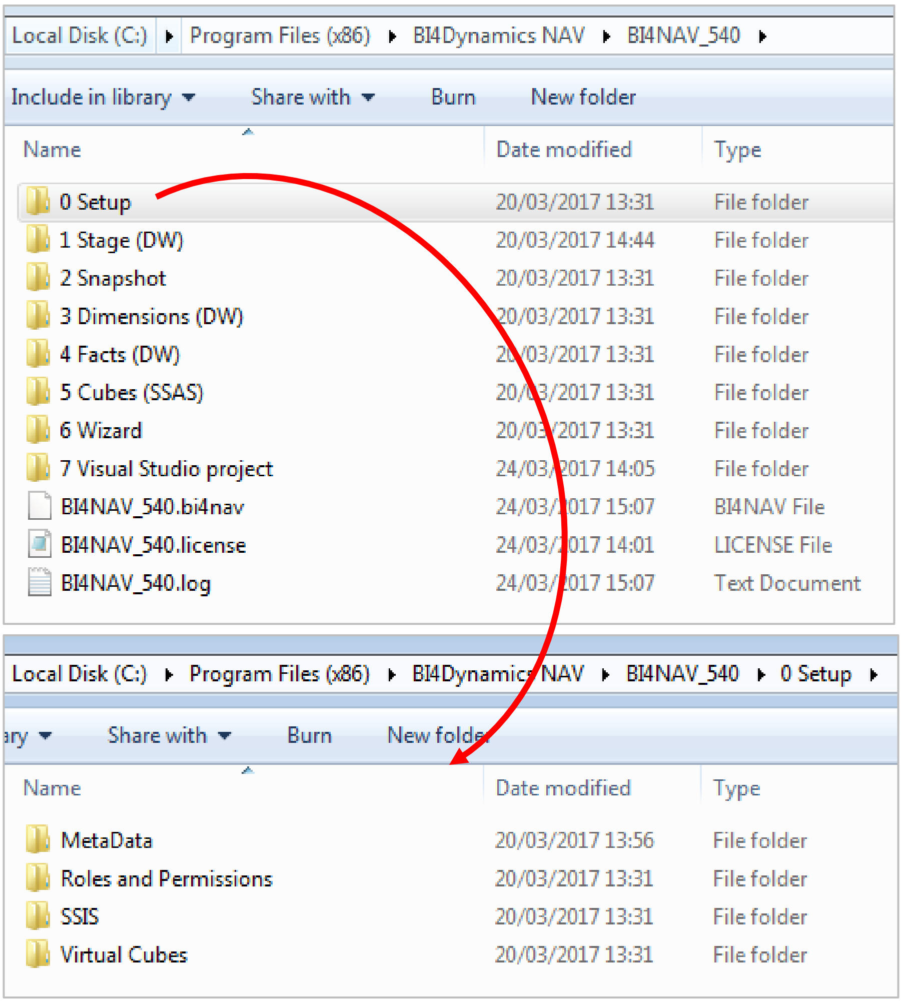
Data Source	TSQL Column Filter Data		
MLADENSSD2 NAV Cronus 2009 Multi-Co...	=0 @3	Save filter	Remove filter
MLADENSSD2\NAVDEMO Demo Database NAV (10-0)		Save filter	Remove filter

12. Run “Deploy & Process” on Current table area button
13. Process DW, Cubes

1.3.2 Add fields to stage

Please refer to our LABS document.

1.4 Setup folder explanation


Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV_540 ▸
Include in library ▾
Share with ▾
Burn
New folder

Name	Date modified	Type
0 Setup	20/03/2017 13:31	File folder
1 Stage (DW)	20/03/2017 14:44	File folder
2 Snapshot	20/03/2017 13:31	File folder
3 Dimensions (DW)	20/03/2017 13:31	File folder
4 Facts (DW)	20/03/2017 13:31	File folder
5 Cubes (SSAS)	20/03/2017 13:31	File folder
6 Wizard	20/03/2017 13:31	File folder
7 Visual Studio project	24/03/2017 14:05	File folder
BI4NAV_540.bi4nav	24/03/2017 15:07	BI4NAV File
BI4NAV_540.license	24/03/2017 14:01	LICENSE File
BI4NAV_540.log	24/03/2017 15:07	Text Document

Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV_540 ▸ 0 Setup ▸
ary ▾
Share with ▾
Burn
New folder

Name	Date modified	Type
MetaData	20/03/2017 13:56	File folder
Roles and Permissions	20/03/2017 13:31	File folder
SSIS	20/03/2017 13:31	File folder
Virtual Cubes	20/03/2017 13:31	File folder

- MetaData (automatic retrieval of NAV Metadata)
- Roles and Permissions (automated saving and applying roles and permissions from analysis cube)
- SSIS (SQL Server Integration Services Processing)
- Virtual Cubes (easily creating a brand new virtual cube)

1.5 MetaData

Information about NAV structures is needed in order BI4Dynamics Wizard to work. We call this set »NAV metadata«. It includes tables, fields, keys, translations, table relations and more.

1.5.1 Metadata - logical explanation

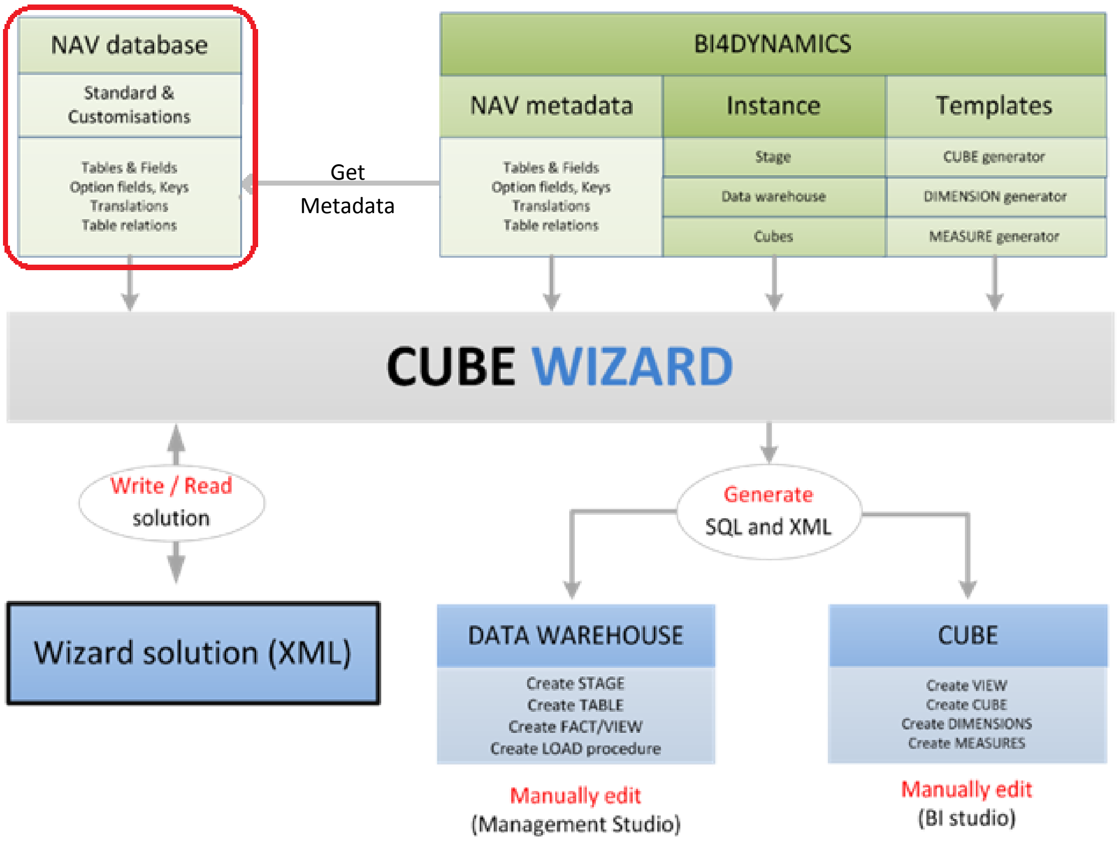
By NAV metadata we understand information about NAV structures that are needed for Wizard. Without this information Wizard would not work.

1.5.2 Metadata - physical explanation

NAV metadata is a NAV table filled by NAV report. Both NAV object are provided by BI4Dynamics. This table is filled by information about NAV structures. These information that are brought to BI part of SQL database and are used by Wizard. Following information are read from NAV:

1. Tables, fields
2. Translation
3. Table keys
4. BI dimension and table relations
5. NAV table relations

All tables and fields (standard and customized) are available.



1.6 Roles and Permissions

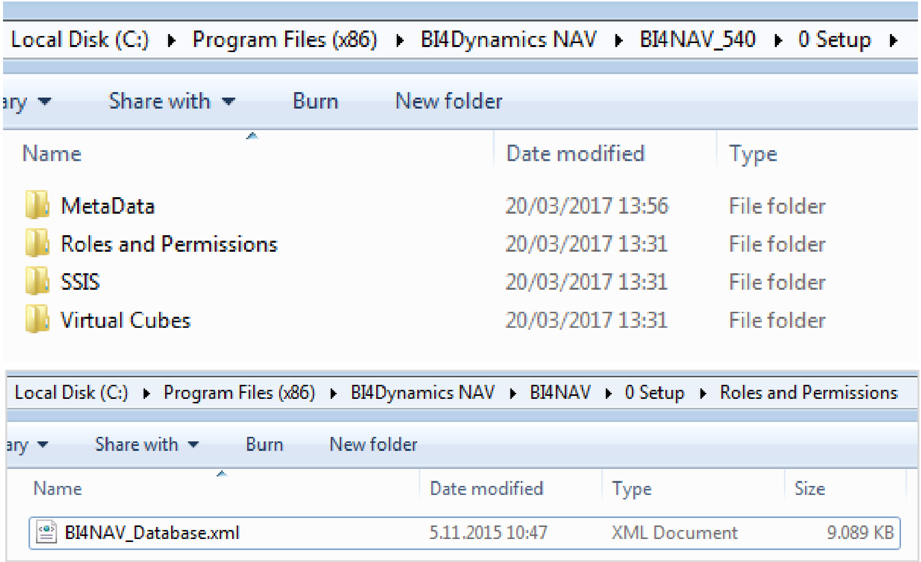
BI4Dynamics version 5.1.0 introduced a new functionality which enable us to keep the security settings on the analysis database intact. In previous versions, the roles which were set up on the analysis database were not restored after Deploy All.

1.6.1 Saving Roles and Permissions

Roles and Permissions are stored to folder automatically:

- before Deploy
- before Processing data

Roles are saved in .xml file in “Roles and Permissions” folder.


Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV_540 ▸ 0 Setup ▸
ary ▾
Share with ▾
Burn
New folder

Name	Date modified	Type
MetaData	20/03/2017 13:56	File folder
Roles and Permissions	20/03/2017 13:31	File folder
SSIS	20/03/2017 13:31	File folder
Virtual Cubes	20/03/2017 13:31	File folder

Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸ 0 Setup ▸ Roles and Permissions
ary ▾
Share with ▾
Burn
New folder

Name	Date modified	Type	Size
BI4NAV_Database.xml	5.11.2015 10:47	XML Document	9.089 KB

1.6.2 Restoring Roles and Permissions

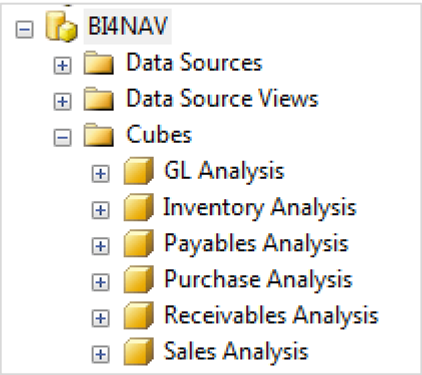
Roles and permissions will be automatically restored to analysis database at the end of each Processing.

1.7 Virtual cubes

Virtual Cubes can be easily created from physical cubes (standard and wizard created cube) by creating a .txt file and saving it to the 0 Setup folder (subfolder Virtual Cubes). Each deploy updates physical and virtual cubes structures, while each process updates the data from NAV.

1.7.1 Initial state

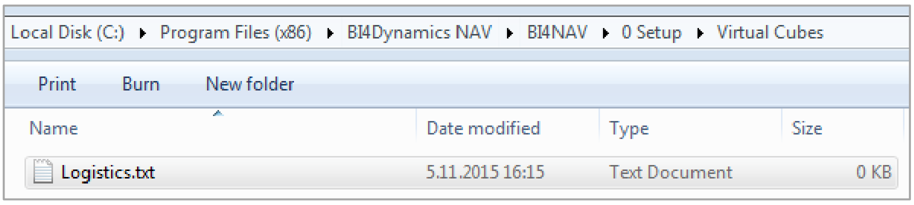
BI4Dynamics standard package cubes are deployed and processed.



View from Management studio

1.7.2 Create virtual cube .txt file

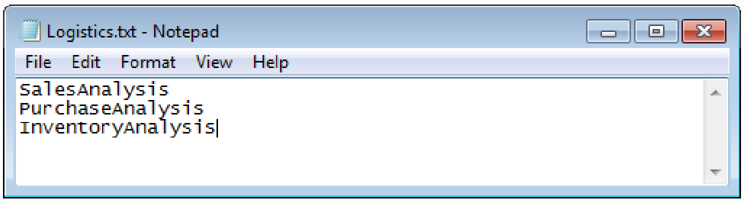
Create a new, empty .txt file and name the same as you would like the new cube to be named. In this scenario, we will create a logistics cube.


Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸ 0 Setup ▸ Virtual Cubes
Print
Burn
New folder

Name	Date modified	Type	Size
Logistics.txt	5.11.2015 16:15	Text Document	0 KB

1.7.3 Which cubes would you like to join?

Just write the name of the cubes in the newly created file (without the spaces). You can join all BI4Dynamics standard package cubes or select just few of them. In our case, we have selected Sales, Purchase and Inventory cubes. Save the file.



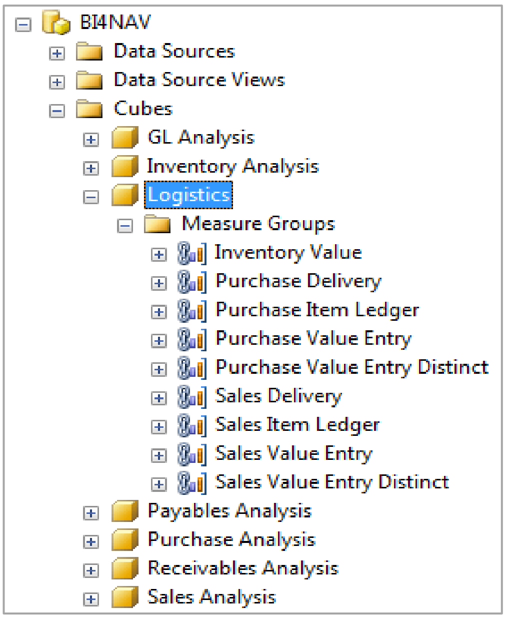
```
SalesAnalysis
PurchaseAnalysis
InventoryAnalysis
```

1.7.4 Deploy and Process

Virtual cubes are deployed and processed after physical cubes. This ensures that all changes (structures and content) in physical cubes will be included in virtual cubes.

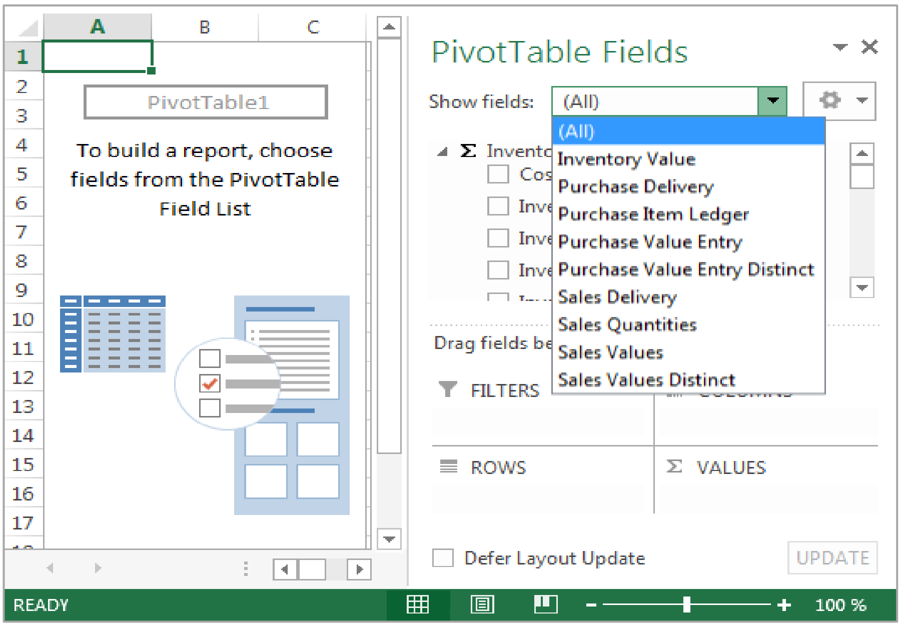
1.7.5 New virtual cube is ready to use!

All measures from physical cubes are available in new virtual cube – Logistics.



- BI4NAV
  - Data Sources
  - Data Source Views
  - Cubes
    - GL Analysis
    - Inventory Analysis
    - Logistics
      - Measure Groups
        - Inventory Value
        - Purchase Delivery
        - Purchase Item Ledger
        - Purchase Value Entry
        - Purchase Value Entry Distinct
        - Sales Delivery
        - Sales Item Ledger
        - Sales Value Entry
        - Sales Value Entry Distinct
    - Payables Analysis
    - Purchase Analysis
    - Receivables Analysis
    - Sales Analysis

View from Management studio



To build a report, choose fields from the PivotTable Field List

Show fields: (All)

- (All)
- Inventory Value
- Purchase Delivery
- Purchase Item Ledger
- Purchase Value Entry
- Purchase Value Entry Distinct
- Sales Delivery
- Sales Quantities
- Sales Values
- Sales Values Distinct

Drag fields be

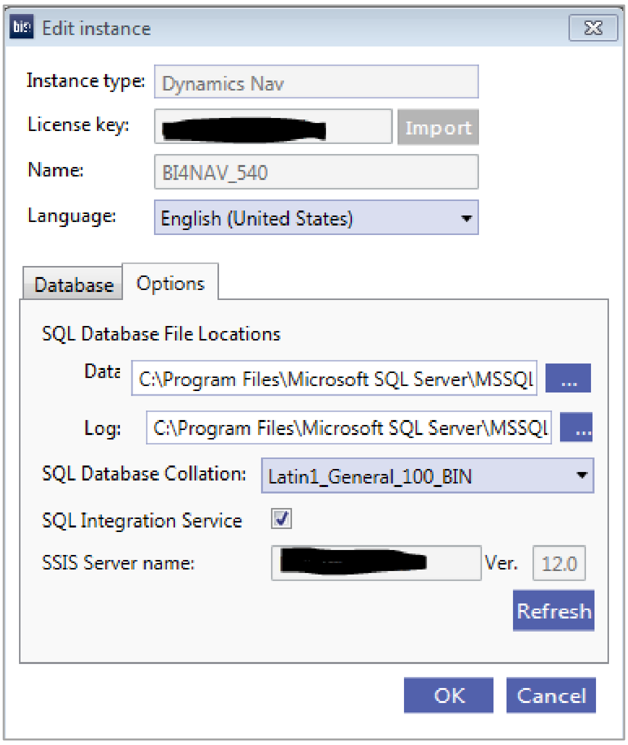
□ Defer Layout Update

The same view from Excel

1.8 SSIS Processing

SQL Server Integration Services processing will process the SSIS packages containing BI4Dynamics stored procedures parallel. If not installed or selected, loading stage tables and processing DW tables will run sequentially, one stored procedures after another.
SQL Server Integration Services processing can be turn on during the creation of the instance or turn on/off from the BI4Dynamics application File menu.

File→Edit→Options→Check/Uncheck SQL Integration Service



Edit instance

Instance type: Dynamics Nav

License key: [redacted] Import

Name: BI4NAV_540

Language: English (United States)

Database | Options

SQL Database File Locations

Data: C:\Program Files\Microsoft SQL Server\MSSQL ...

Log: C:\Program Files\Microsoft SQL Server\MSSQL ...

SQL Database Collation: Latin1_General_100_BIN

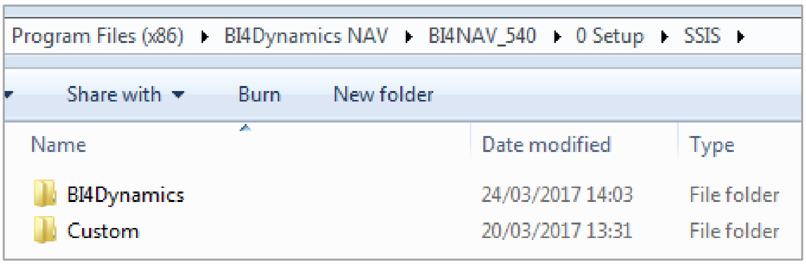
SQL Integration Service ☑

SSIS Server name: [redacted] Ver. 12.0

Refresh

OK Cancel

1.8.1 SSIS



Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV_540 ▸ 0 Setup ▸ SSIS ▸

Name	Date modified	Type
BI4Dynamics	24/03/2017 14:03	File folder
Custom	20/03/2017 13:31	File folder

BI4Dynamics folder contains 21 encrypted, non-editable SSIS packages for BI4Dynamics troubleshooting. Please contact support@bi4dynamics.com if errors occur on processing using SSIS option.

A bran new feature is being developed allowing the custom SSIS packages to be included in BI4Dynamics process flow.

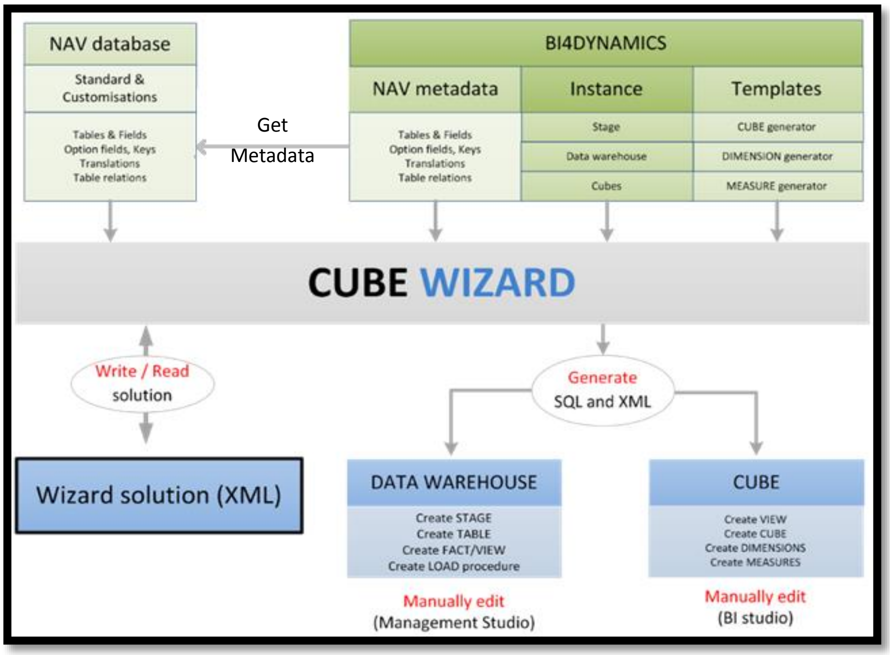
2 WIZARD

2.1 Wizard features

BI4Dynamics NAV Cube Wizard can:

- Manage cubes
    - Create new cubes
    - Add fact to existing cubes (standard or wizard generated)
- Manage dimensions
    - Create new dimensions
    - Change existing dimensions
- Manage measures
    - Extend all standard measures with YTD, Year-Over-Year, Last Periods (12), Period-Over-Period set of measures
    - Add custom MDX function
- Manage BI solution
    - Work with BI4Dynamics standard, Wizard and Microsoft tools simultaneously
    - Write T-SQL scripts with SQL Server Management Studio
    - Write XML files (dim, cube) with BIDS (Business Intelligence Development Studio)

Cubes that are generated by Wizard can be used in any language (that is installed in computer that is running NAV metadata report).



- Wizard does not require any SQL knowledge

2.1.1 Manage measures

Measures are created in different levels:

1 DW based measures

- Simple data warehouse based measures: SUM, MIN, MAX

2 Extended measures

DW based measure functionality can be extended. These are predefined measure groups that are optionally generated by Wizard:

- YTD (Year-To-Date) – 5 measures,
- POP (Period-Over-Period) – 2 measures
- YOY (Year-Over-Year) – 2 measures,
- LAST PERIODS (one measure for each of last 12 months),
- ROLLING (Rolling 12 m and index) – 2 measures.

3 Custom measures

All custom MDX is supported and any measure based on MDX can be added to any cube.

2.1.2 Manage dimensions

1 Dimensions

- Dimensions are created directly from NAV fields. Supported are following field types:
  - Code,
  - Text,
  - Boolean,
  - Option fields,
  - Integer;
- Dimensions are created from NAV fields with table relation:
  - Simple table relations,
  - Complex table relations (2 levels – se examples below)

14. BI4Dynamics supports 99% of all table relations in NAV.
15.

Example 1 : T17/F10 Balancing Account = Option field + Related table

- T18 (Customer), T25 (Vendor), T5600 (Fixed Asset), T270 (Bank Account)

Example 2 : T37/F10 Sales Line No

- T27 (Item), T25 (GL Account), T5600 (Fixed Asset), T270 (Bank Account

2 Dimension attributes

- All fields are added as dimension attributes to DW and cube. Flow fields and flow filters are not included.
- Wizard can create only those attributes that have values in NAV (when using HasValue (option True) in NAV metadata report – if not all attributes are shown); Option, Boolean Field Types are always shown.

3 Dimension hierarchies

- All tables in table relation with dimension table are added as dimension hierarchies

Created are only hierarchies for those tables that have values in NAV (when using HasValue (option True) in NAV metadata report – if not all hierarchies are shown);

2.1.3 Manage cubes - create a new cube or add a fact to existing cube

Wizard is currently supporting following functionality:

1 Create cubes

Cubes are created from NAV source table.

- Fact can be created from one source table; usually this is a transactional table (ledger entry, document line or similar)
- One cube can have one or more facts.

2 Add fact to any cube

By adding fact user will add new measures and new dimension from any NAV source table to standard functionality; this feature does not change standard functionality (and cause possible errors) but expands it; Facts can be added to any existing cubes (standard or wizard generated).

3 Manage Physical and Role-Playing dimensions

There is no need to create similar physical dimension more than once. One physical dimension can be create more times using same structure and different name. Such dimension are called Role–Playing dimensions. Wizard can create and re-use physical and Role-Playing dimension.

Example 1 :

- Physical dimension = Customer
- Role-playing dimension: Bill-To Customer, Sell-To Customer

Example 2 :

- Physical dimension = Date
- Role-playing dimension: Date, Shipment Date, Document Date, Order Date

4 Multi-measure tool

Multi-measure tool dimension is added automatically. Multi-measure tool will be added when:

1. One date dimension is selected.
2. More than one dimension is selected and Dimension name of either one is “Date”.

2.1.4 Manage BI solution - work simultaneously with BI4Dynamics and Microsoft tools

System and manually generated code coexist. Wizard generates SQL and XML code into predefined folders of instance. These scripts can be further used and modified by Microsoft Management Studio or Visual Studio.

1 Manage T-SQL scripts with Management Studio

Any existing scripts can be modified and saved to instance solution folder. Depending on the functionality (stage tables, snap-shot database, dimensions and facts) – different folder structure is used. These modified scripts are executed after standard scripts and therefore always overwrite standard solution.

2 Manage XML with BIDS

Solution created by BIDS (Business intelligence Development Studio) can be saved to BI4Dynamics folder, (Full solution folder or part of solution - only one dim XML file) deployed and processed just like cubes created by Wizard. Following process is supported:

1. Create cube with standard deploy process or with Wizard
2. Open and modify cube with BIDS
3. Save BIDS solution to folder (5-Develpment)
4. Deploy and process instance with BI4Dynamics

There is very simple rule which code will be executed – custom or manually changed:

- Code that is last in folder structure will be executed by framework.
- Previous versions of code will be overwritten

User can continually use BI4Dynamics standard or Wizard and manually generated code by Microsoft tools. This is not likely with some ISV solutions, that after working with Microsoft tools, ISV solution can no longer be used.

3 Manage Roles and Permissions

Roles and Permission are created within standard Microsoft OLAP cubes (SSAS). BI4Dynamics Roles and Permissions are stored and revoked automatically on deployment.

4 Language features

- Every language available in NAV database is also available in cubes created by Wizard.
- Translations are available for all dimensions and for measures that are generated directly from fields.
- Translation of dimension names, measures and facts (measure groups) can be changed.

2.1.5 Distribution of Wizard solution file

This feature enables easy distribution of BI solution or BI project among different projects. Partner who develops BI vertical solution once can distribute solution as a XML file. Wizard will be able to read solution and generate SQL and XML scripts based on NAV metadata from new instance.

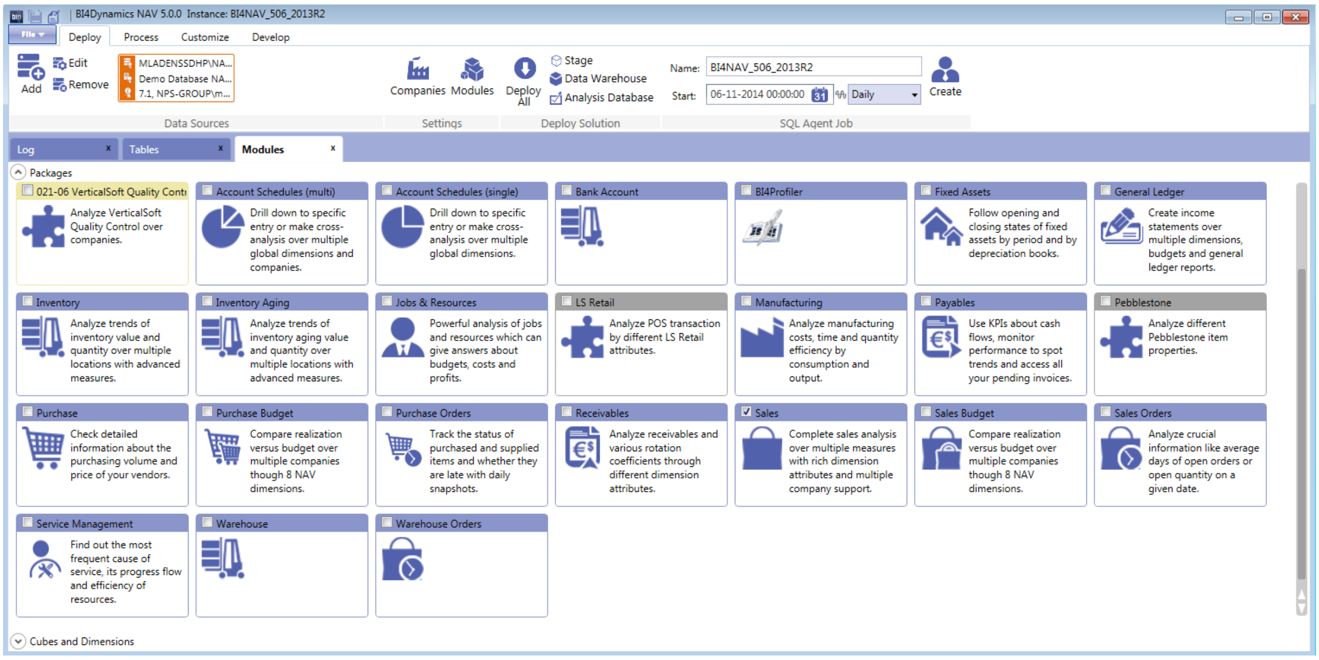
Wizard may work also when NAV structures (NAV metadata) in source and target installation are not the same, but must have same NAV languages!

2.2 Preparation

This section shows how to prepare instance for Wizard.

2.2.1 Create BI4Dynamics instance

First you need to install BI4Dynamics version 5 which includes the Wizard. Follow the installation instructions of BI4Dynamics NAV Installation Manual. After installing BI4Dynamics you have to deploy and process at least one BI4Dynamics module.



BI4Dynamics NAV 5.0.0 Instance: BI4NAV_506_2013R2

MLADENSSDHP\NA...
Demo Database NA...
7.1, NPS-GROUP\m...

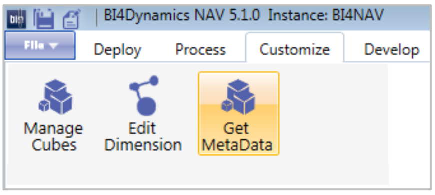
Name: BI4NAV_506_2013R2

Start: 06-11-2014 00:00:00 Daily

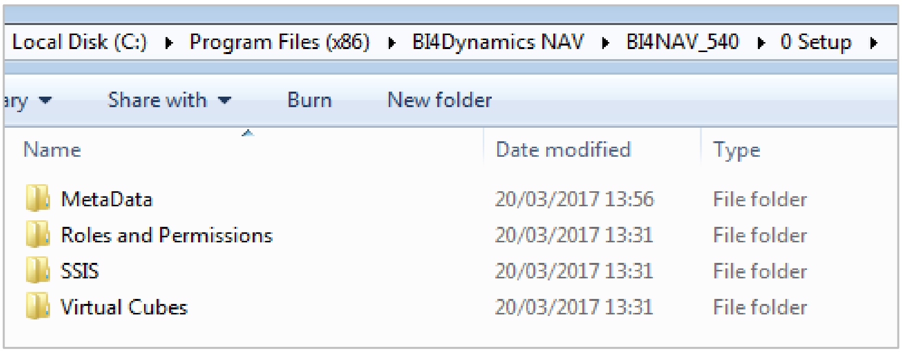
- ☐ 021-06 VerticalSoft Quality Contr: Analyze VerticalSoft Quality Control over companies.
- ☐ Account Schedules (multi): Drill down to specific entry or make cross-analysis over multiple global dimensions and companies.
- ☐ Account Schedules (single): Drill down to specific entry or make cross-analysis over multiple global dimensions.
- ☐ Bank Account
- ☐ BI4Profiler
- ☐ Fixed Assets: Follow opening and closing states of fixed assets by period and by depreciation books.
- ☐ General Ledger: Create income statements over multiple dimensions, budgets and general ledger reports.
- ☐ Inventory: Analyze trends of inventory value and quantity over multiple locations with advanced measures.
- ☐ Inventory Aging: Analyze trends of inventory aging value and quantity over multiple locations with advanced measures.
- ☐ Jobs & Resources: Powerful analysis of jobs and resources which can give answers about budgets, costs and profits.
- ☐ LS Retail: Analyze POS transaction by different LS Retail attributes.
- ☐ Manufacturing: Analyze manufacturing costs, time and quantity efficiency by consumption and output.
- ☐ Payables: Use KPIs about cash flows, monitor performance to spot trends and access all your pending invoices.
- ☐ Pebblestone: Analyze different Pebblestone item properties.
- ☐ Purchase: Check detailed information about the purchasing volume and price of your vendors.
- ☐ Purchase Budget: Compare realization versus budget over multiple companies though 8 NAV dimensions.
- ☐ Purchase Orders: Track the status of purchased and supplied items and whether they are late with daily snapshots.
- ☐ Receivables: Analyze receivables and various rotation coefficients through different dimension attributes.
- ☑ Sales: Complete sales analysis over multiple measures with rich dimension attributes and multiple company support.
- ☐ Sales Budget: Compare realization versus budget over multiple companies though 8 NAV dimensions.
- ☐ Sales Orders: Analyze crucial information like average days of open orders or open quantity on a given date.
- ☐ Service Management: Find out the most frequent cause of service, its progress flow and efficiency of resources.
- ☐ Warehouse
- ☐ Warehouse Orders

2.2.2 Get NAV Metadata

Once you have successfully deployed and processed at least one BI4Dynamics module, **MetaData are retrieved automatically**. If some changes is made in Microsoft Dynamics NAV, manual retrieval of Get MetaData is possible by pressing Get MetaData.



MetaData will be stored in folder “0 Setup” in Bi4Dynamics instance folder.

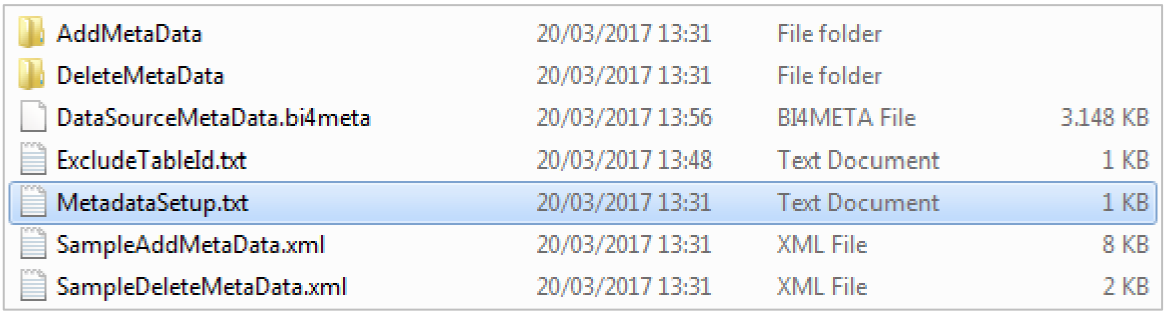

Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV_540 ▸ 0 Setup ▸
ary
Share with
Burn
New folder

Name	Date modified	Type
MetaData	20/03/2017 13:56	File folder
Roles and Permissions	20/03/2017 13:31	File folder
SSIS	20/03/2017 13:31	File folder
Virtual Cubes	20/03/2017 13:31	File folder

1 MetaData Setup

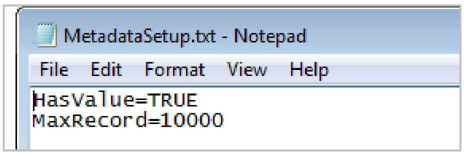
Metadata folder contains two folders and five files.

One is the .BI4META file where all the data is stored. The others are setup files to adjust the properties of the metadata.



AddMetaData	20/03/2017 13:31	File folder	
DeleteMetaData	20/03/2017 13:31	File folder	
DataSourceMetaData.bi4meta	20/03/2017 13:56	BI4META File	3.148 KB
ExcludeTableId.txt	20/03/2017 13:48	Text Document	1 KB
MetadataSetup.txt	20/03/2017 13:31	Text Document	1 KB
SampleAddMetaData.xml	20/03/2017 13:31	XML File	8 KB
SampleDeleteMetaData.xml	20/03/2017 13:31	XML File	2 KB

Setup file:

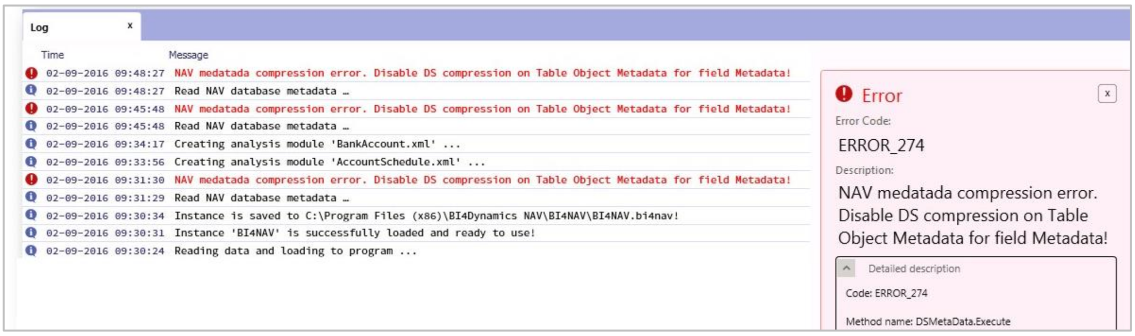


```
HasValue=TRUE
MaxRecord=10000
```

If HasValue is set to FALSE, all the fields from NAV tables will be imported and saved to the BI4META file.
If HasValue is set to TRUE, only the fields which contains values will be imported to the BI4META file.

When HasValue is set to TRUE, another parameter is there to help us import the Metadata quicker. When MaxRecord is set to 10.000, only the tables with less than 10.000 records will be scanned for field values. All the fields will be automatically saved to Metadata when a specific table has more than 10.000 records.
This value can be adjusted within the .txt to fit the needs of a specific customer.

2 Troubleshooting



```
Time                 Message
02-09-2016 09:48:27  NAV medatada compression error. Disable DS compression on Table Object Metadata for field Metadata!
02-09-2016 09:48:27  Read NAV database metadata …
02-09-2016 09:45:48  NAV medatada compression error. Disable DS compression on Table Object Metadata for field Metadata!
02-09-2016 09:45:48  Read NAV database metadata …
02-09-2016 09:34:17  Creating analysis module 'BankAccount.xml' ...
02-09-2016 09:33:56  Creating analysis module 'AccountSchedule.xml' ...
02-09-2016 09:31:30  NAV medatada compression error. Disable DS compression on Table Object Metadata for field Metadata!
02-09-2016 09:31:29  Read NAV database metadata …
02-09-2016 09:30:34  Instance is saved to C:\Program Files (x86)\BI4Dynamics NAV\BI4NAV\BI4NAV.bi4nav!
02-09-2016 09:30:31  Instance 'BI4NAV' is successfully loaded and ready to use!
02-09-2016 09:30:24  Reading data and loading to program ...
```

Error

Error Code:
ERROR_274

Description:
NAV medatada compression error. Disable DS compression on Table Object Metadata for field Metadata!

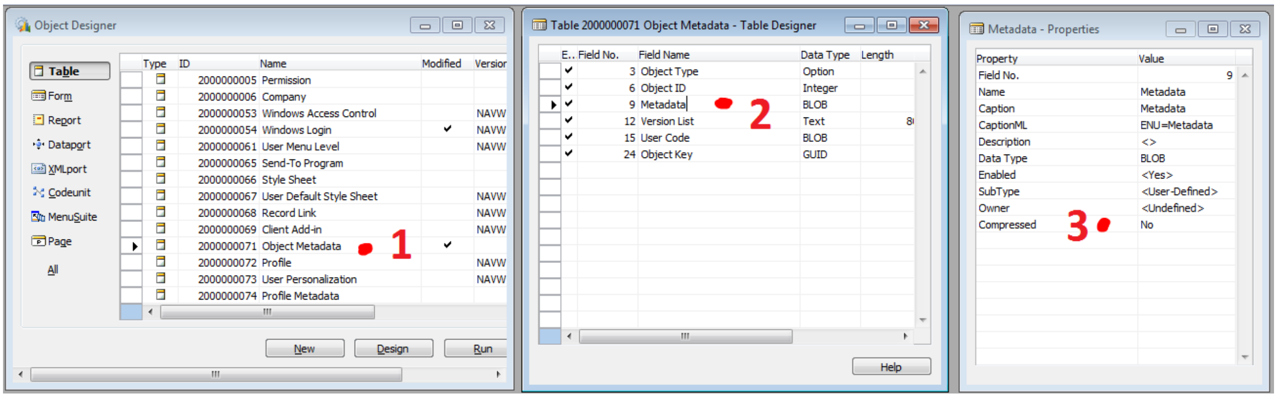
Detailed description

Code: ERROR_274

Method name: DSMetaData.Execute

This error can happen only in NAV 2009. To fix it you have to open NAV database, go to Object Designer and select:

- Table Object Metadata
- Field Metadata
- set property **Compressed** = **No** (by default is set to **Yes**)



Type	ID	Name	Modified	Versior
	2000000005	Permission		
	2000000006	Company		
	2000000053	Windows Access Control		NAVW
	2000000054	Windows Login	✓	NAVW
	2000000061	User Menu Level		NAVW
	2000000065	Send-To Program		
	2000000066	Style Sheet		
	2000000067	User Default Style Sheet		NAVW
	2000000068	Record Link		NAVW
	2000000069	Client Add-in		NAVW
	2000000071	Object Metadata	✓	
	2000000072	Profile		NAVW
	2000000073	User Personalization		NAVW
	2000000074	Profile Metadata		

E..	Field No.	Field Name	Data Type	Length
✓	3	Object Type	Option	
✓	6	Object ID	Integer	
✓	9	Metadata	BLOB	
✓	12	Version List	Text	8
✓	15	User Code	BLOB	
✓	24	Object Key	GUID	

Property	Value
Field No.	9
Name	Metadata
Caption	Metadata
CaptionML	ENU=Metadata
Description	<>
Data Type	BLOB
Enabled	<Yes>
SubType	<User-Defined>
Owner	<Undefined>
Compressed	No

In higher version this is already set by default.

2.3 Create cube with wizard

It is recommend to add one cube, deploy instance and add another cube to have all dimensions from previous installation process available in new cubes.

2.3.1 Step 1: Manage cube

1 Go to “Customize” tab

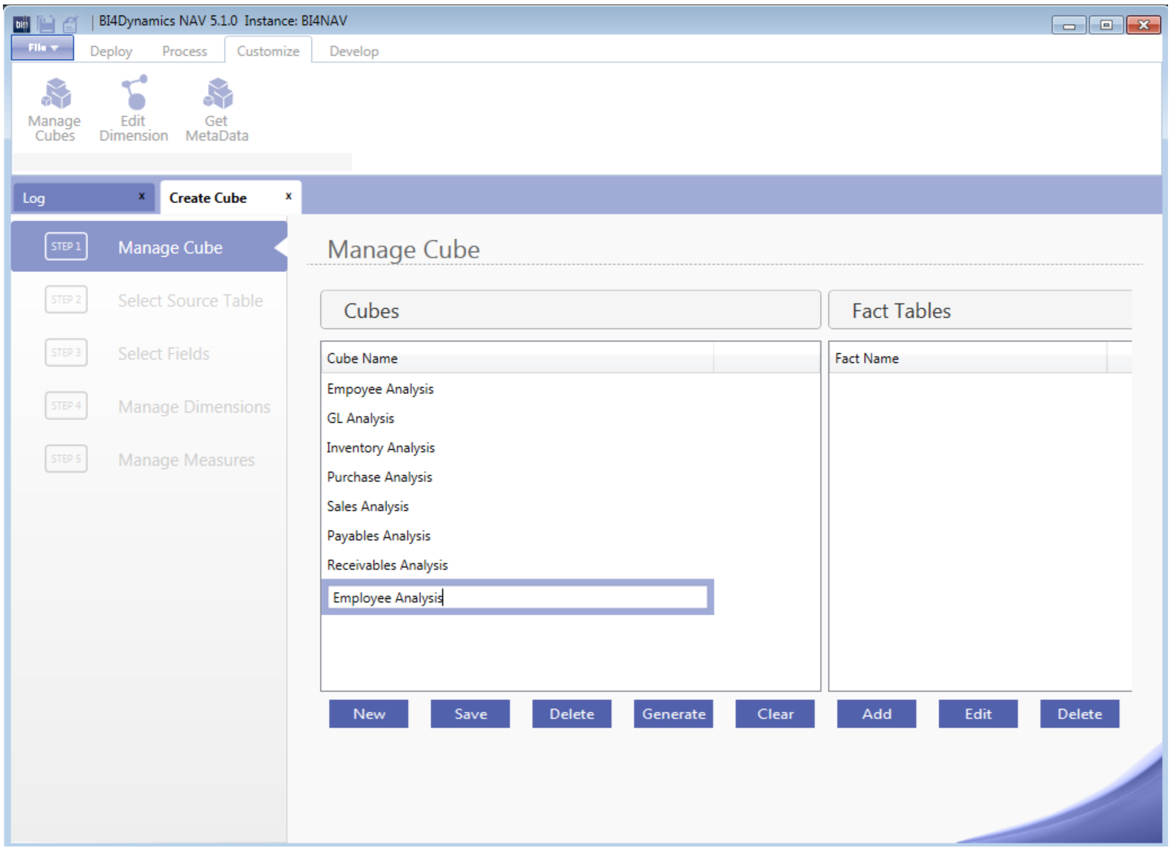
2 Click on “Get MetaData” – wait for Metadata to load.

3 Click on “Manage Cube”

4 Click on “New”

5 Type in the name of new cube

Normally you want to create a new cube type in unique name. If you will type in one of BI4Dynamics existing Cube you will overwrite it.



Cube Name
Empoyee Analysis
GL Analysis
Inventory Analysis
Purchase Analysis
Sales Analysis
Payables Analysis
Receivables Analysis
Employee Analysis

Fact Name

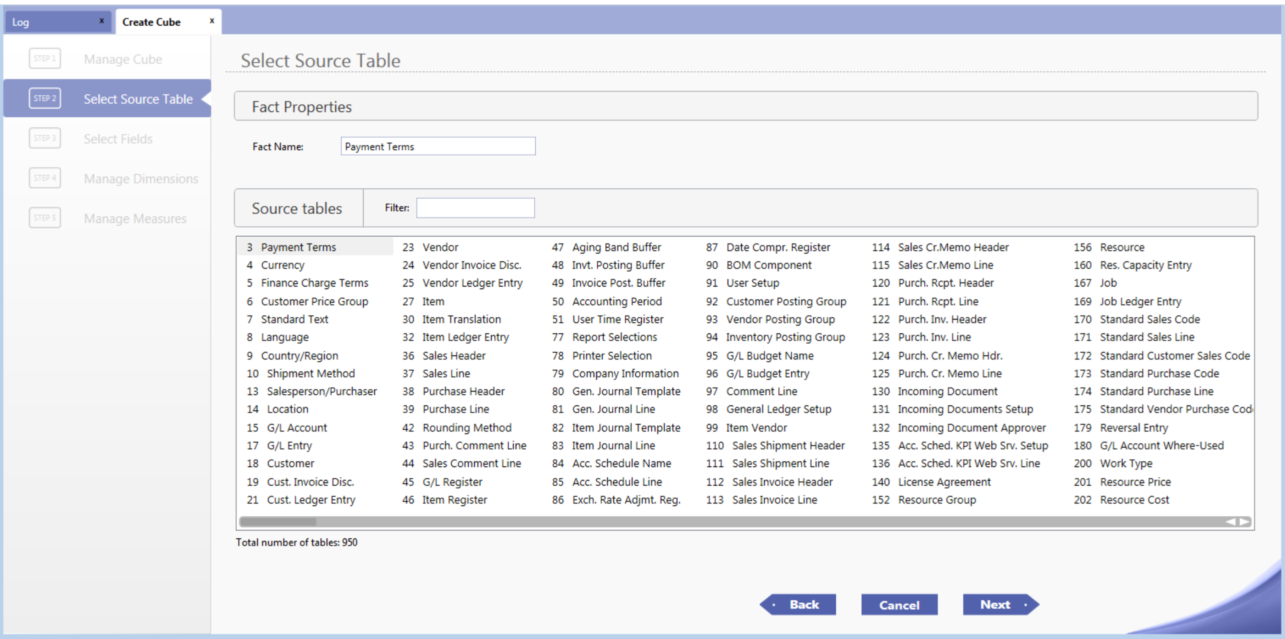
Example of cube name: Employee Analysis

6 Click “Add”

We have added cube and wizard will go automatically to next step.

2.3.2 Step 2: Select source table

New step will open window showing all available NAV tables. Usually there are 900+ tables, depends on NAV version. If you have a vertical solution or custom tables, all will be available. First NAV table “Payment Terms” is always offered as first source table.



Fact Name: Payment Terms

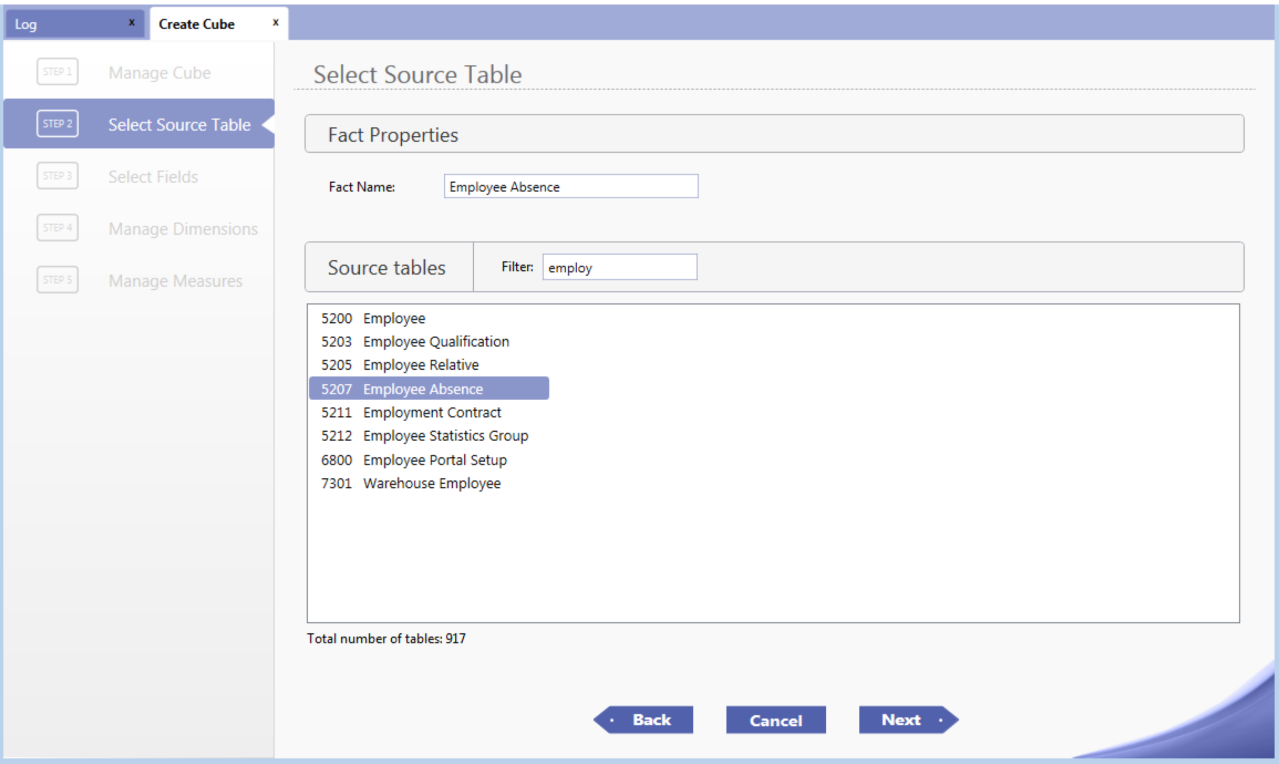
Filter:

3 Payment Terms
4 Currency
5 Finance Charge Terms
6 Customer Price Group
7 Standard Text
8 Language
9 Country/Region
10 Shipment Method
13 Salesperson/Purchaser
14 Location
15 G/L Account
17 G/L Entry
18 Customer
19 Cust. Invoice Disc.
21 Cust. Ledger Entry
23 Vendor
24 Vendor Invoice Disc.
25 Vendor Ledger Entry
27 Item
30 Item Translation
32 Item Ledger Entry
36 Sales Header
37 Sales Line
38 Purchase Header
39 Purchase Line
42 Rounding Method
43 Purch. Comment Line
44 Sales Comment Line
45 G/L Register
46 Item Register
47 Aging Band Buffer
48 Invt. Posting Buffer
49 Invoice Post. Buffer
50 Accounting Period
51 User Time Register
77 Report Selections
78 Printer Selection
79 Company Information
80 Gen. Journal Template
81 Gen. Journal Line
82 Item Journal Template
83 Item Journal Line
84 Acc. Schedule Name
85 Acc. Schedule Line
86 Exch. Rate Adjmt. Reg.
87 Date Compr. Register
90 BOM Component
91 User Setup
92 Customer Posting Group
93 Vendor Posting Group
94 Inventory Posting Group
95 G/L Budget Name
96 G/L Budget Entry
97 Comment Line
98 General Ledger Setup
99 Item Vendor
110 Sales Shipment Header
111 Sales Shipment Line
112 Sales Invoice Header
113 Sales Invoice Line
114 Sales Cr.Memo Header
115 Sales Cr.Memo Line
120 Purch. Rcpt. Header
121 Purch. Rcpt. Line
122 Purch. Inv. Header
123 Purch. Inv. Line
124 Purch. Cr. Memo Hdr.
125 Purch. Cr. Memo Line
130 Incoming Document
131 Incoming Documents Setup
132 Incoming Document Approver
135 Acc. Sched. KPI Web Srv. Setup
136 Acc. Sched. KPI Web Srv. Line
140 License Agreement
152 Resource Group
156 Resource
160 Res. Capacity Entry
167 Job
169 Job Ledger Entry
170 Standard Sales Code
171 Standard Sales Line
172 Standard Customer Sales Code
173 Standard Purchase Code
174 Standard Purchase Line
175 Standard Vendor Purchase Cod
179 Reversal Entry
180 G/L Account Where-Used
200 Work Type
201 Resource Price
202 Resource Cost

Total number of tables: 950

1 Use filter to find source table

16. We have entered “employ” to shortlist 7 tables and selected “Employee Absence”. This will be also name of the fact.



Fact Name: Employee Absence

Filter: employ

5200 Employee
5203 Employee Qualification
5205 Employee Relative
5207 Employee Absence
5211 Employment Contract
5212 Employee Statistics Group
6800 Employee Portal Setup
7301 Warehouse Employee

Total number of tables: 917

2 Click “Next” to move to next step

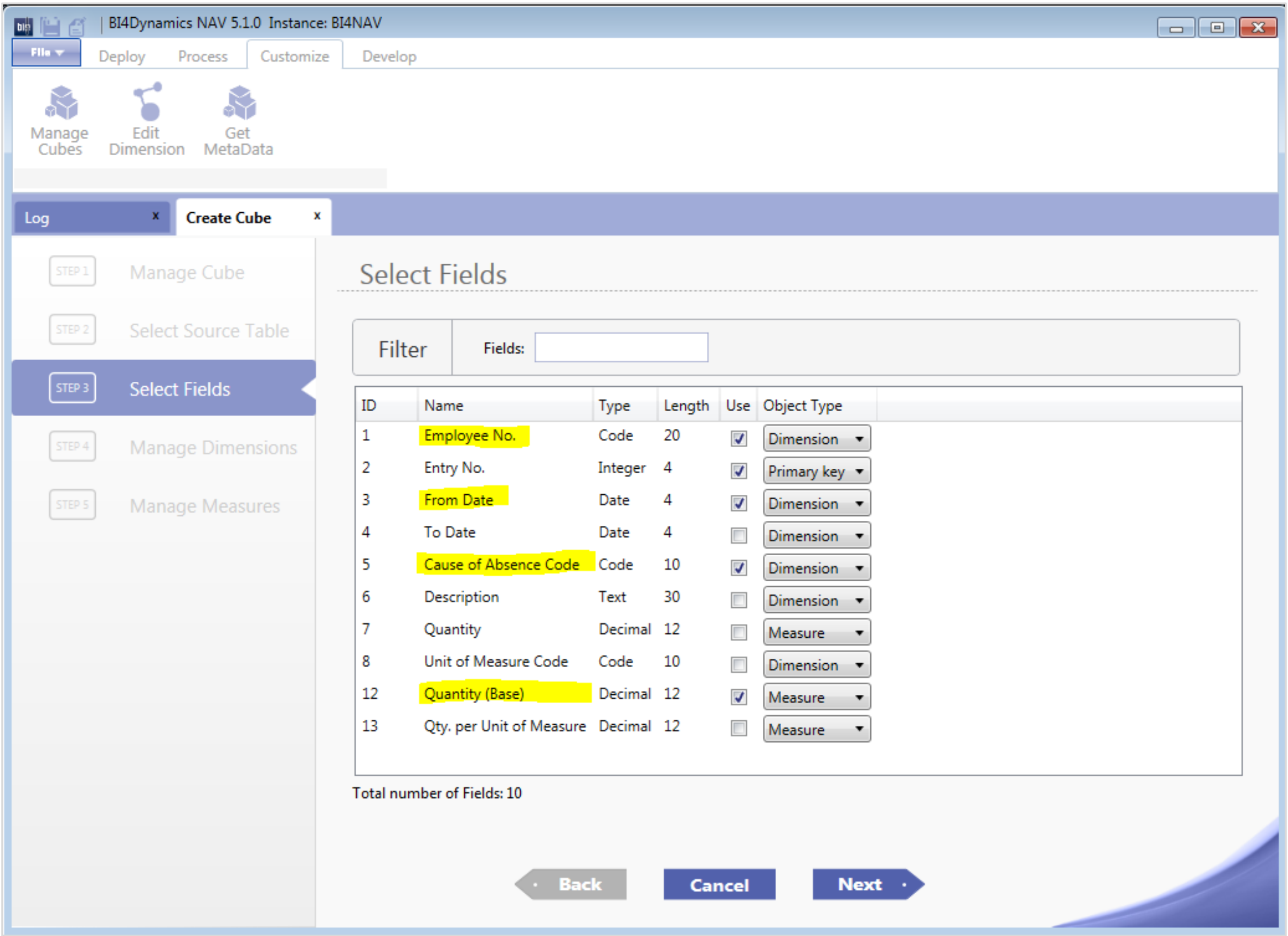
2.3.3 Step 3: Select fields

Wizard automatically defines Object Type.

- Primary key: a primary key in NAV
- Dimension: for fields Code, Date or Text
- Measures: Decimal, Integer

1 Check selected fields

Don't forget to include at least one date dimension. You cannot create a Dimension with Decimal Type. Better to leave suggested Object Type.



Select Fields

Filter Fields:

ID	Name	Type	Length	Use	Object Type
1	Employee No.	Code	20	☑	Dimension
2	Entry No.	Integer	4	☑	Primary key
3	From Date	Date	4	☑	Dimension
4	To Date	Date	4	□	Dimension
5	Cause of Absence Code	Code	10	☑	Dimension
6	Description	Text	30	□	Dimension
7	Quantity	Decimal	12	□	Measure
8	Unit of Measure Code	Code	10	□	Dimension
12	Quantity (Base)	Decimal	12	☑	Measure
13	Qty. per Unit of Measure	Decimal	12	□	Measure

Total number of Fields: 10

2 Click "Next" to move to next step

If you want to use primary key as dimension Change Object type to Dimension. This is not used often as most transactional tables have integer as primary key. Example: it can be used if table "Job Planning line" is used and Job is a part of primary key. If we don't change Object Type in row Job, Job dimension will not be available.

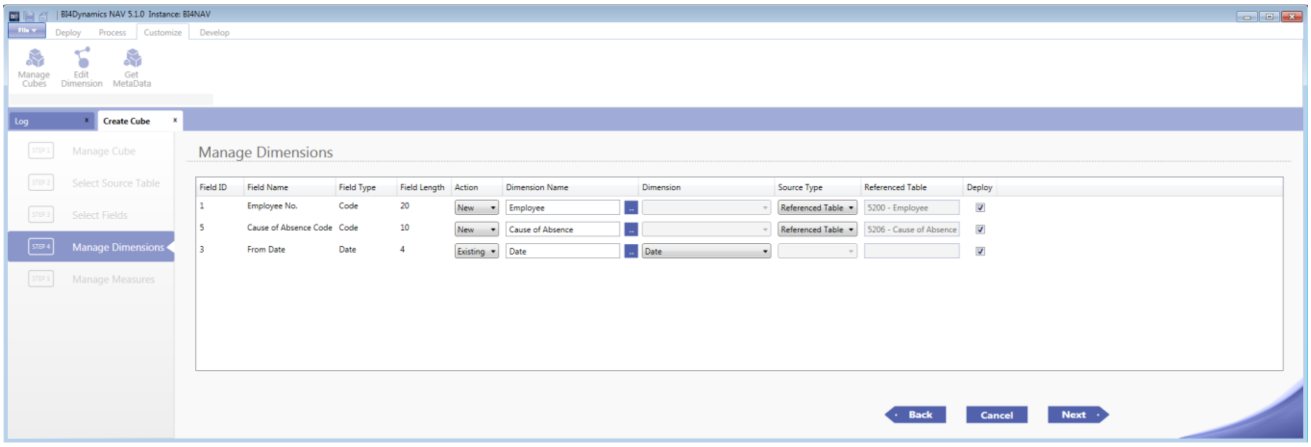
2.3.4 Step 4: Manage dimensions

This is a step where fields are transformed into dimensions. In most cases user will have little work in this step as NAV metadata model and BI templates are brought together to create the right dimension. Following are BI fields in this step:

1 Action

There are 2 action types:

- **New**: a new dimension will be created from this field.
- **Existing**: field is connected with existing dimensions and existing dimension will be brought to this cube



Manage Dimensions

Field ID	Field Name	Field Type	Field Length	Action	Dimension Name	Dimension	Source Type	Referenced Table	Deploy
1	Employee No.	Code	20	New	Employee		Referenced Table	5200 - Employee	☑
5	Cause of Absence Code	Code	10	New	Cause of Absence		Referenced Table	5206 - Cause of Absence	☑
3	From Date	Date	4	Existing	Date	Date			☑

2 Dimension name

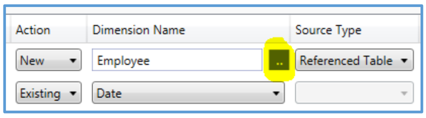
3 Select new Dimension

If wizard didn't find relation between field and any NAV table (NAV metadata hasn't been updated or there is no relation written for this table) than new dimension can be selected.

This field can be changed if new dimension is created and we would like to change the name.

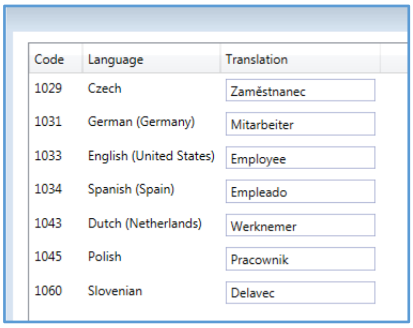
4 Change translations

At the end of this field is short blue section with dots.



Action	Dimension Name	Source Type
New	Employee ..	Referenced Table
Existing	Date	

By clicking here user can change translations:



Code	Language	Translation
1029	Czech	Zaměstnanec
1031	German (Germany)	Mitarbeiter
1033	English (United States)	Employee
1034	Spanish (Spain)	Empleado
1043	Dutch (Netherlands)	Werknemer
1045	Polish	Pracownik
1060	Slovenian	Delavec

5 Check existing Dimension

List of existing dimension comes from:

- Dimension from standard instance (modules that have been implemented in this instance)

- Dimension from Cubes created by wizard; these dimensions are available if Cube created by wizard has been deployed.

6 Source Type

Source of dimension can be:

- Reference table
- Current field

7 Dimension name

It is possible to connect same dimension to different fields more than once. Dimension name must be change so that name is unique. By this process we create Role-Playing dimensions.

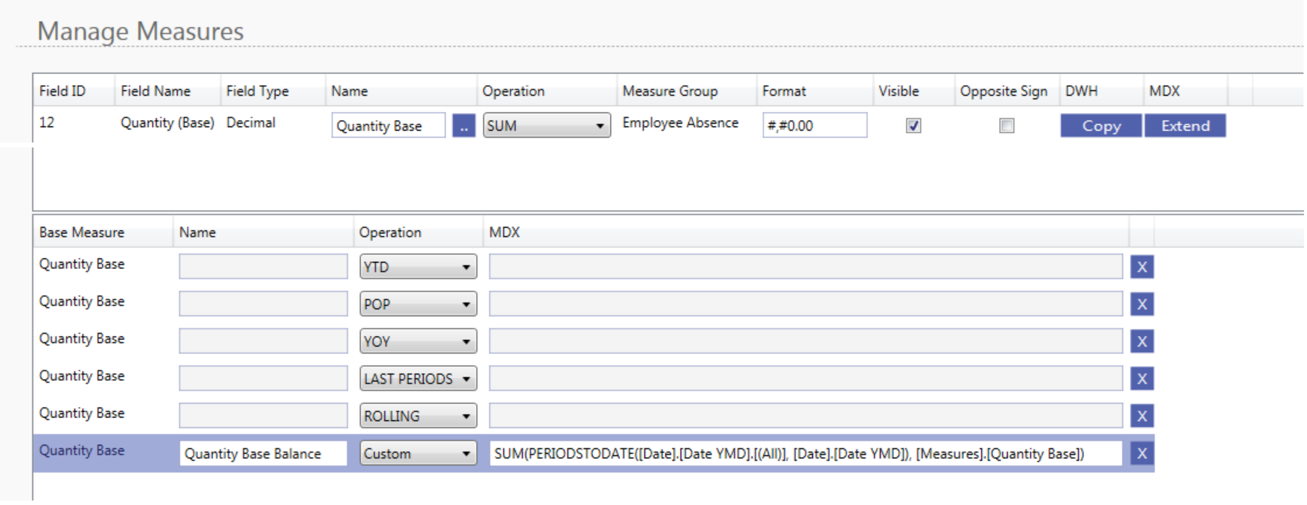
> - Example 1: Source table T37 (Sales line), Posting date, Shipment date
>
> Date dimension is suggested for both fields (Posting date and Shipment date). User should change name for Shipment date.
>
> - Example 2: T37 (Sales line), Bill-to Customer, Sell-to Customer
>
> Both fields can be connected to existing dimension Customer

8 Deploy

By default this field is checked, meaning, that field is used and SQL scripts will be generated. If we uncheck this field no SQL script will be generated.

Click next to proceed to following step.

2.3.5 Step 5: Manage measures



Manage Measures

Field ID	Field Name	Field Type	Name	Operation	Measure Group	Format	Visible	Opposite Sign	DWH	MDX
12	Quantity (Base)	Decimal	Quantity Base ..	SUM	Employee Absence	#,#0.00	☑	☐	Copy	Extend

Base Measure	Name	Operation	MDX	
Quantity Base		YTD		X
Quantity Base		POP		X
Quantity Base		YOY		X
Quantity Base		LAST PERIODS		X
Quantity Base		ROLLING		X
Quantity Base	Quantity Base Balance	Custom	SUM(PERIODSTODATE([Date].[Date YMD].[(All)], [Date].[Date YMD]), [Measures].[Quantity Base])	X

This is an example of how user can extend the usage of measure "Quantity base":

This is description of fields in this step:

1 Name

Every measure needs to have a unique name. Language translations from NAV are available and can be changed.

2 Operation

Available are operations supported by standard SQL: SUM (most common), MIN, MAX, COUNT and DISTINC COUNT;

3 Measure group

Name of fact from step 2 is name of measure group that will group measures generated in this step.

4 Format

Propose is most common decimal format.

5 Visible

By default all measures are visible. If you would use a measure for MDX calculations in other measures and not need to show it in cubes, that you would uncheck this field.

6 Opposite sign

Check this field if you want to change a sign.

7 Action COPY

Add additional measures with different operations. Change Name and Operation. This measure are based on DW and can also be extended in next step.

8 Action EXTEND

Measure from this line is copied to window below where additional operations can be added. One measure can be copied more times. Write in the Name and change Operation.

Add predefined measure groups:

- YTD add 5 measures
- LAST PERIODS adds 12 measures

Add custom MDX scripts

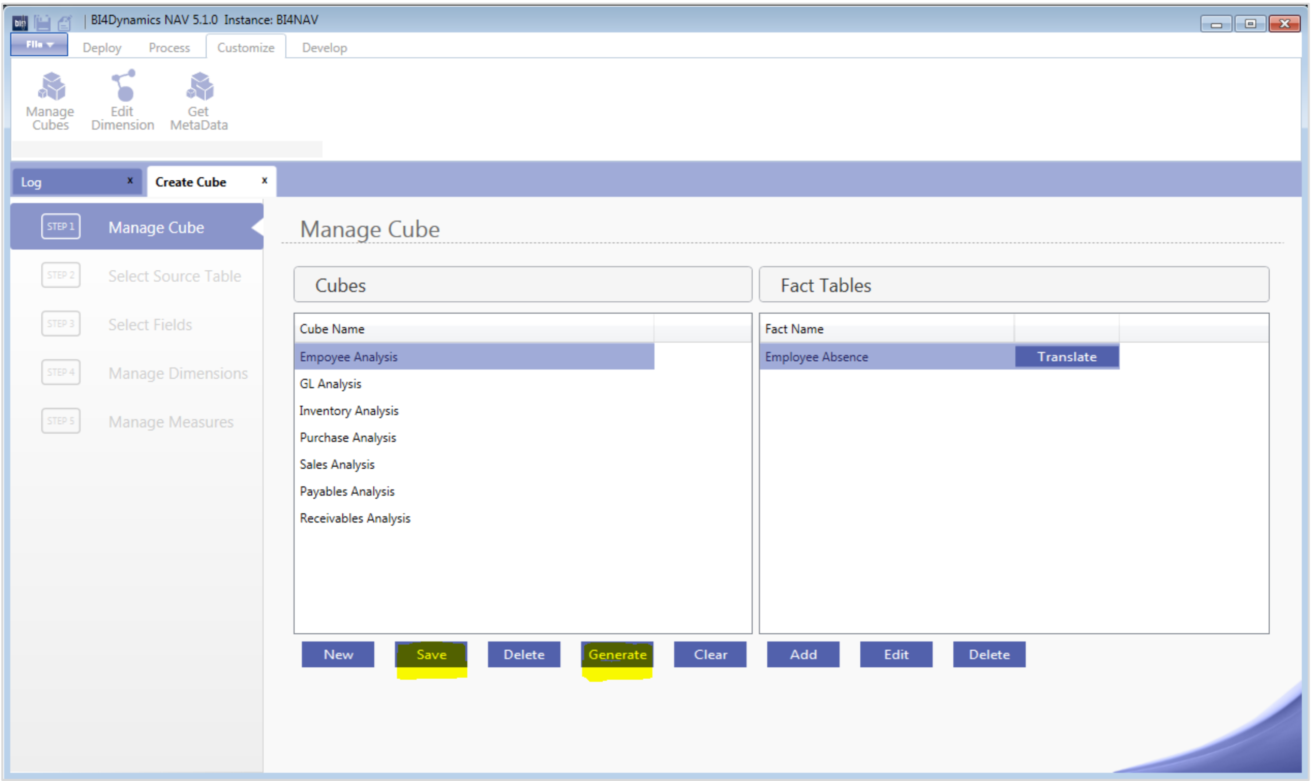
By selecting option Custom any MDX script can be added. Wizard will not check if MDX is correct. This has to be done (if needed) outside wizard in SQL tools.

9 Next

By selecting “Next” changes are written in memory.

2.3.6 Save solution and generate scripts

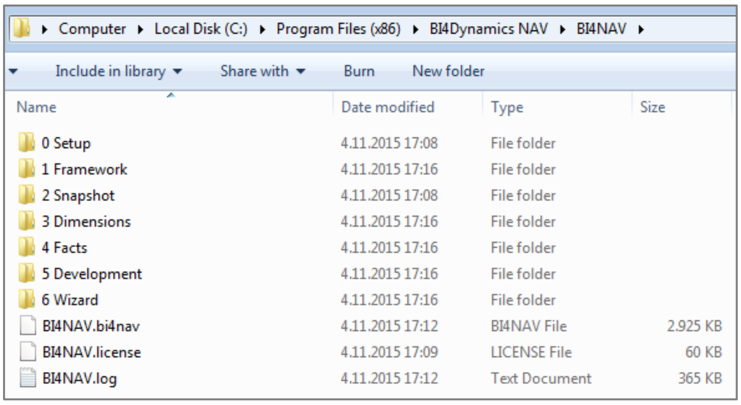
Wizard is at step 1, where we can save wizard solution and generate scripts.



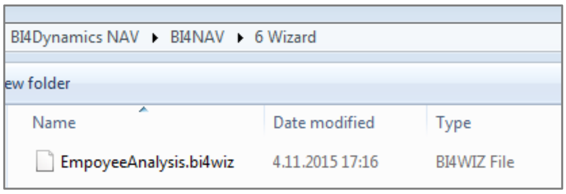
Cube Name
Empoyee Analysis
GL Analysis
Inventory Analysis
Purchase Analysis
Sales Analysis
Payables Analysis
Receivables Analysis

Fact Name
Employee Absence

1 Save


Computer ▸ Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸
Include in library
Share with
Burn
New folder

Name	Date modified	Type	Size
0 Setup	4.11.2015 17:08	File folder	
1 Framework	4.11.2015 17:16	File folder	
2 Snapshot	4.11.2015 17:08	File folder	
3 Dimensions	4.11.2015 17:16	File folder	
4 Facts	4.11.2015 17:16	File folder	
5 Development	4.11.2015 17:16	File folder	
6 Wizard	4.11.2015 17:16	File folder	
BI4NAV.bi4nav	4.11.2015 17:12	BI4NAV File	2.925 KB
BI4NAV.license	4.11.2015 17:09	LICENSE File	60 KB
BI4NAV.log	4.11.2015 17:12	Text Document	365 KB



Name	Date modified	Type
EmpoyeeAnalysis.bi4wiz	4.11.2015 17:16	BI4WIZ File

Wizard's solution is saved to instance folder "6-Wizard" with name of cube.

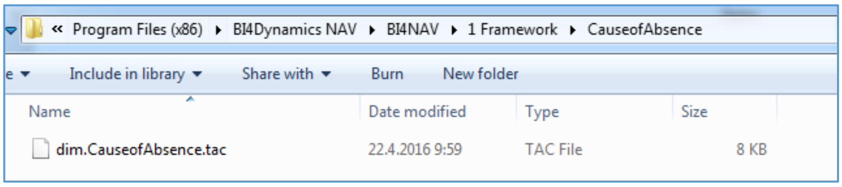
2 Generate

Wizard will create scripts to instance folder.

- Scripts are executed during deploy and process after standard part of BI4Dynamics
- Scripts can be manually changed; we suggest to copy them to specific folder (out of cube folder)
- Scripts inside BI4Dynamics folder can be delete by deleting cube or fact; scripts in every custom folder have to be deleted manually.

Following scripts are generated (not all are shown here):

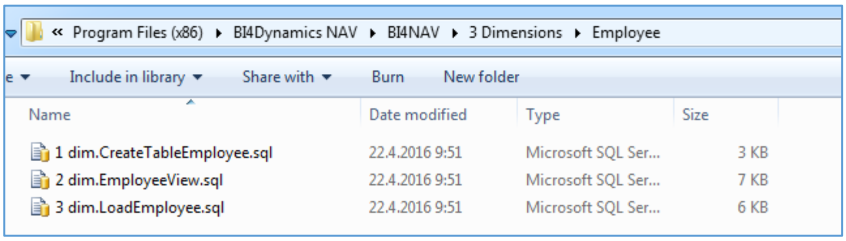
Stage files



« Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸ 1 Framework ▸ CauseofAbsence

Name	Date modified	Type	Size
dim.CauseofAbsence.tac	22.4.2016 9:59	TAC File	8 KB

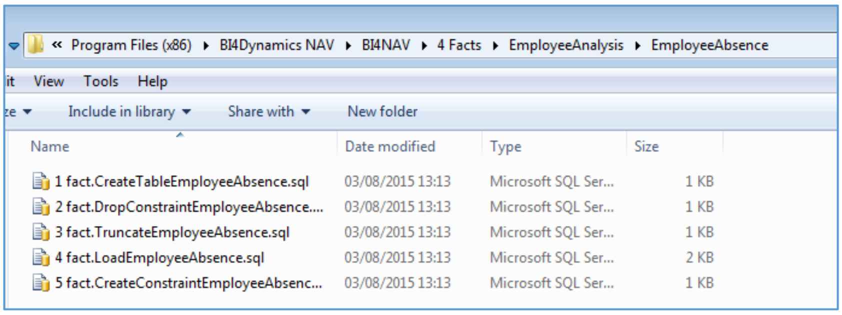
Dimension files



« Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸ 3 Dimensions ▸ Employee

Name	Date modified	Type	Size
1 dim.CreateTableEmployee.sql	22.4.2016 9:51	Microsoft SQL Ser...	3 KB
2 dim.EmployeeView.sql	22.4.2016 9:51	Microsoft SQL Ser...	7 KB
3 dim.LoadEmployee.sql	22.4.2016 9:51	Microsoft SQL Ser...	6 KB

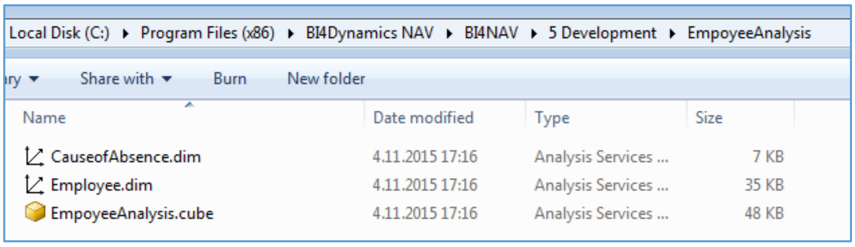
Fact files



« Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸ 4 Facts ▸ EmployeeAnalysis ▸ EmployeeAbsence

Name	Date modified	Type	Size
1 fact.CreateTableEmployeeAbsence.sql	03/08/2015 13:13	Microsoft SQL Ser...	1 KB
2 fact.DropConstraintEmployeeAbsence....	03/08/2015 13:13	Microsoft SQL Ser...	1 KB
3 fact.TruncateEmployeeAbsence.sql	03/08/2015 13:13	Microsoft SQL Ser...	1 KB
4 fact.LoadEmployeeAbsence.sql	03/08/2015 13:13	Microsoft SQL Ser...	2 KB
5 fact.CreateConstraintEmployeeAbsenc...	03/08/2015 13:13	Microsoft SQL Ser...	1 KB

Cube files



Local Disk (C:) ▸ Program Files (x86) ▸ BI4Dynamics NAV ▸ BI4NAV ▸ 5 Development ▸ EmpoyeeAnalysis

Name	Date modified	Type	Size
CauseofAbsence.dim	4.11.2015 17:16	Analysis Services ...	7 KB
Employee.dim	4.11.2015 17:16	Analysis Services ...	35 KB
EmpoyeeAnalysis.cube	4.11.2015 17:16	Analysis Services ...	48 KB

3 Deploy and process

1. Click x (close the Create Cube tab)
2. Go to Deploy Tab -> Deploy All
3. Wait for successful deployment
4. Go to Process Tab -> Process All

After these steps all cubes (standard and wizard generated) are available for analysis.

2.4 MANAGE WIZARD GENERATED CUBE

2.4.1 Edit existing fact from Wizard Cube

We can edit cube or fact by clicking Edit. Don't forget to save changes and generate scripts.

2.4.2 Add new fact to Wizard Cube

New facts can be added to existing cubes, regardless weather they are standard or generated by Wizard;

1 Manage (re-use) cube dimension

It is possible to use dimension from previous facts. Dimensions that have been created in previous facts have to be deployed to be available in new facts

2 Manage measures

Set unique names

It is important to set unique names for measures.

Example:

- In previous fact we have used Job ledger entry where field Quantity is available.
- In current fact we use Job planning line where also field Quantity is available.

We should put name Planned quantity in second fact to avoid errors. Also not forget to change translations if needed.

Calculate any measure

In MDX section of extended measures user can use all measures that have been generated in this cube form previous facts.

2.4.3 Copy cube from another instance

Wizard solution file from folder 6-Wizard can be coped to any BI4Dynamics instance with same BI4Dynamics version and same (or at least similar) NAV structure. When we open Wizard all files that are in folder "6-Wizard" are presented as cubes.

2.5 MANAGE EXISTING STANDARD CUBE

2.5.1 Add new fact to existing standard BI4Dynamics cube

New facts can be added to any standard BI4Dynamics cubes; Process is same as Adding fact to Wizard Cube.

2.5.2 Edit existing fact

When editing fact that has been added to standard cube in previous process, program will generate error on generate, because measure group is still in cube. User needs to delete cube by deploying clean script for cube first and then generate scripts.

1 Action to generate error

1. Add fact to existing cube
2. Save cube, generate scripts, deploy & process cube
3. Edit fact
4. Save cube
5. Error on Generate scripts

2 Error message

Measure group with name "Fact_name_entry" already exists

3 Correct process

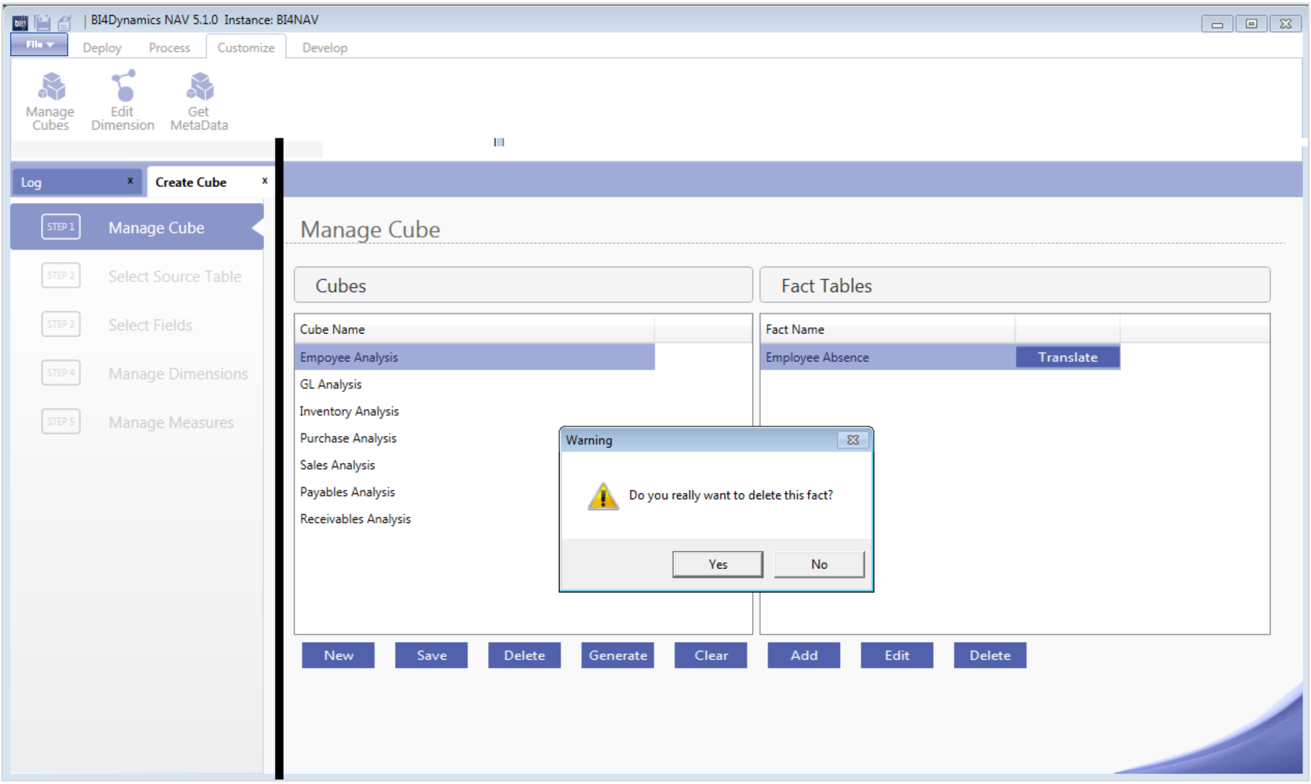
1. Save cube, generate scripts, deploy & process cube
2. DELETE FOLDER 5 WITH FACT NAME
3. DEPLOY CUBE
4. Edit fact
5. Save cube
6. Generate scripts (no error)

4 Explanation

Solution reads cube structure on open Customize \ Manage cube. Overwrite of wizard generated fact is not allowed. Fact must be removed from cube by deploying cube without fact in order to be inserted (edited) again.

2.5.3 Delete Wizard generated fact that has been added to standard cube

User has deleted fact from cube but fact is still there because cube has not been saved after delete.



Cube Name
Empoyee Analysis
GL Analysis
Inventory Analysis
Purchase Analysis
Sales Analysis
Payables Analysis
Receivables Analysis

Fact Name
Employee Absence

Do you really want to delete this fact?

1 Action to generate error

1. Delete fact that has been added to standard cube
2. Open Customize \ Manage Cube again
3. Deleted fact is still there

2 Correct process

1. Delete fact that has been added to standard cube
2. SAVE CUBE
3. Open Customize \ Manage Cube again
4. Deleted fact not there anymore

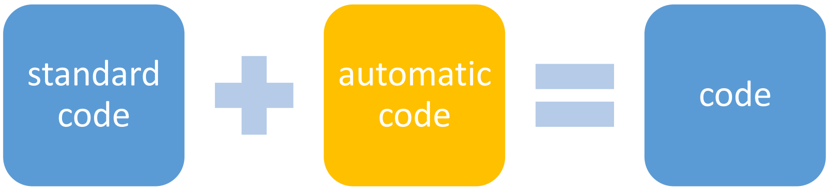
3 Explanation

Delete fact deletes all files in instance folder. Information about fact is still written in file “CUBE_NAME.bi4wiz” in folder 6 until one of following steps is done:

a) Cube is saved and only current (updated) structures (without deleted fact) are written to cube
b) Delete file from folder 6

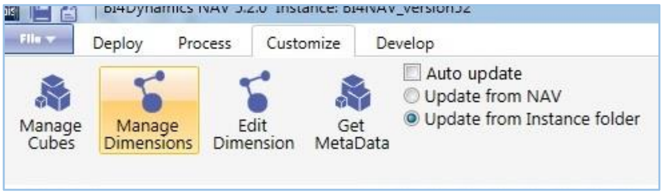
2.6 MANAGE EXISTING DIMENSIONS

2.6.1 Introduction to Manage Dimensions Feature


standard code + automatic code = code

BI4Dynamics version 5.2.0 is bringing new automatization functionality to dimension behavior based on attributes usage each process time with Manage Dimensions Feature.
It enables you to extend existing dimensions of simple table relation with automatically (wizard) generated attributes and hierarchies.

2.6.2 Prerequisites


BI4Dynamics NAV 5.2.0 Instance: BI4NAV_version52

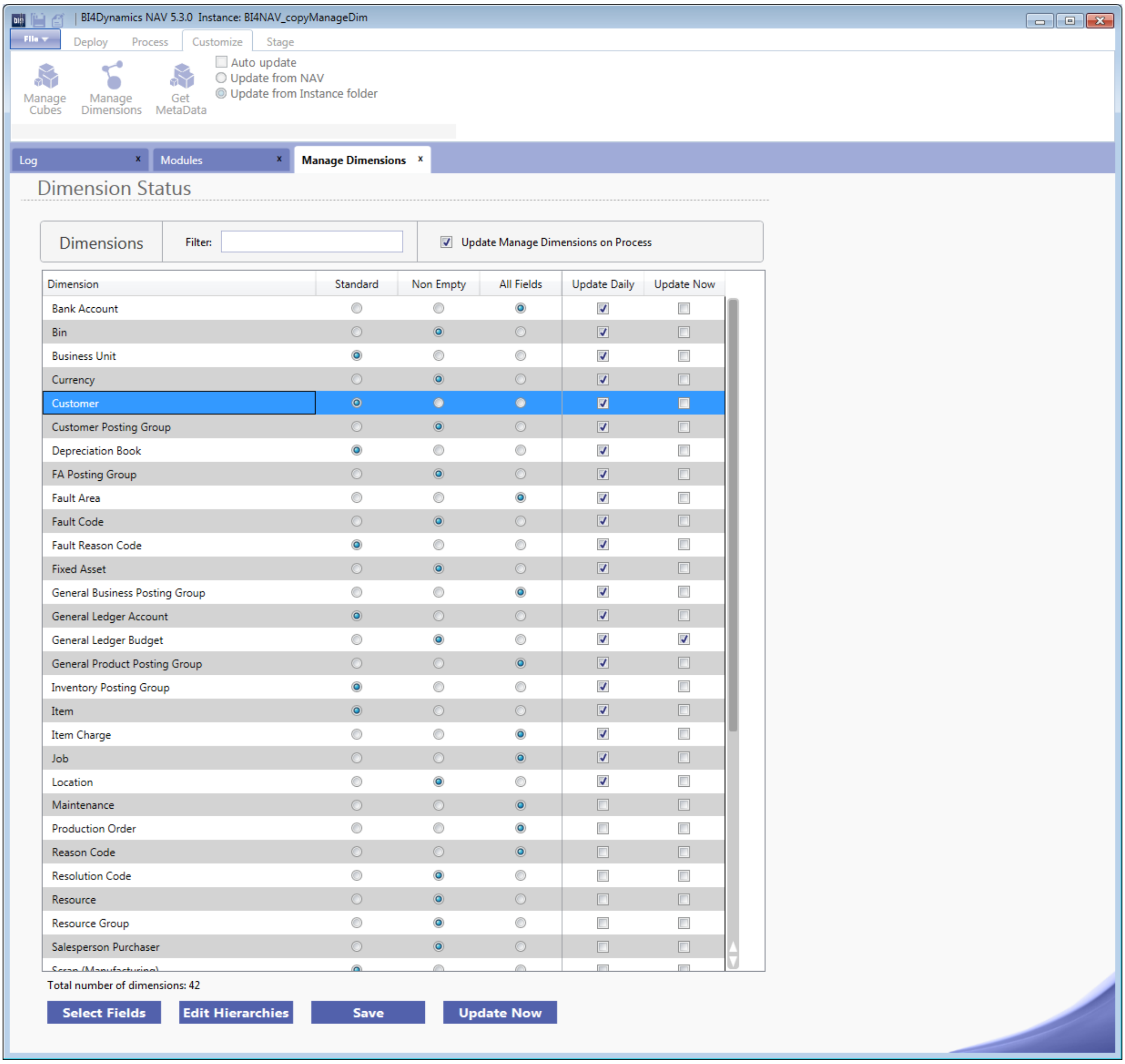
□ Auto update
○ Update from NAV
● Update from Instance folder

Before getting into Manage Dimensions menu following must be provided:

- Solution deployed (all areas),
- Metadata imported.

2.6.3 Dimensions Setup

Dimensions properties can be managed through **Manage Dimensions** menu. You can select dimension's status between 3 options – by default all standard dimensions are in **Standard** state, which can be extended with **Non Empty** or **All Fields**.
Dimension's processing can be executed straight from application. All dimensions with **Update Now** selected are being updated after clicking on **Update Now** button.
Dimensions updating can also be enabled through each **Process All** execution – this process can be executed through application or Agent Job. Enabled main property **Update Manage Dimensions on Process** will update all dimensions with **Update Daily** action selected.



□ Auto update
○ Update from NAV
● Update from Instance folder

Dimension Status

Dimensions Filter: ☑ Update Manage Dimensions on Process

Dimension	Standard	Non Empty	All Fields	Update Daily	Update Now
Bank Account	○	○	●	☑	□
Bin	○	●	○	☑	□
Business Unit	●	○	○	☑	□
Currency	○	●	○	☑	□
Customer	●	○	○	☑	□
Customer Posting Group	○	●	○	☑	□
Depreciation Book	●	○	○	☑	□
FA Posting Group	○	●	○	☑	□
Fault Area	○	○	●	☑	□
Fault Code	○	●	○	☑	□
Fault Reason Code	●	○	○	☑	□
Fixed Asset	○	●	○	☑	□
General Business Posting Group	○	○	●	☑	□
General Ledger Account	●	○	○	☑	□
General Ledger Budget	○	●	○	☑	☑
General Product Posting Group	○	○	●	☑	□
Inventory Posting Group	●	○	○	☑	□
Item	●	○	○	☑	□
Item Charge	○	○	●	☑	□
Job	○	○	●	☑	□
Location	○	●	○	☑	□
Maintenance	○	○	●	□	□
Production Order	○	○	●	□	□
Reason Code	○	○	●	□	□
Resolution Code	○	●	○	□	□
Resource	○	●	○	□	□
Resource Group	○	●	○	□	□
Salesperson Purchaser	○	●	○	□	□
Scrap (Manufacturing)	●	○	○	□	□

Total number of dimensions: 42

2.6.4 Generated file location

Each dimension execution is leaded through three areas (stage, data warehouse, analytical). Scripts are created with Wizard script generator, stored separately into 3 different folders:

Area	File	Folder	File name
Stage	Tables and columns file	***1 Framework/(DimensionName)***	**dim.*(DimensionName)*.tac** XML file with new fields and related tables
Data warehouse	SQL scripts	***3 Dimensions/(DimensionName)***	**1 dim.CreateTable*(DimensionName)*.sql** **2 dim.*(DimensionName)*View.sql** **3 dim.Load*(DimensionName)*.sql**
Analytical	Analytical dimension file	***5 Development/(DimensionName)***	***(DimensionName)*.dim**

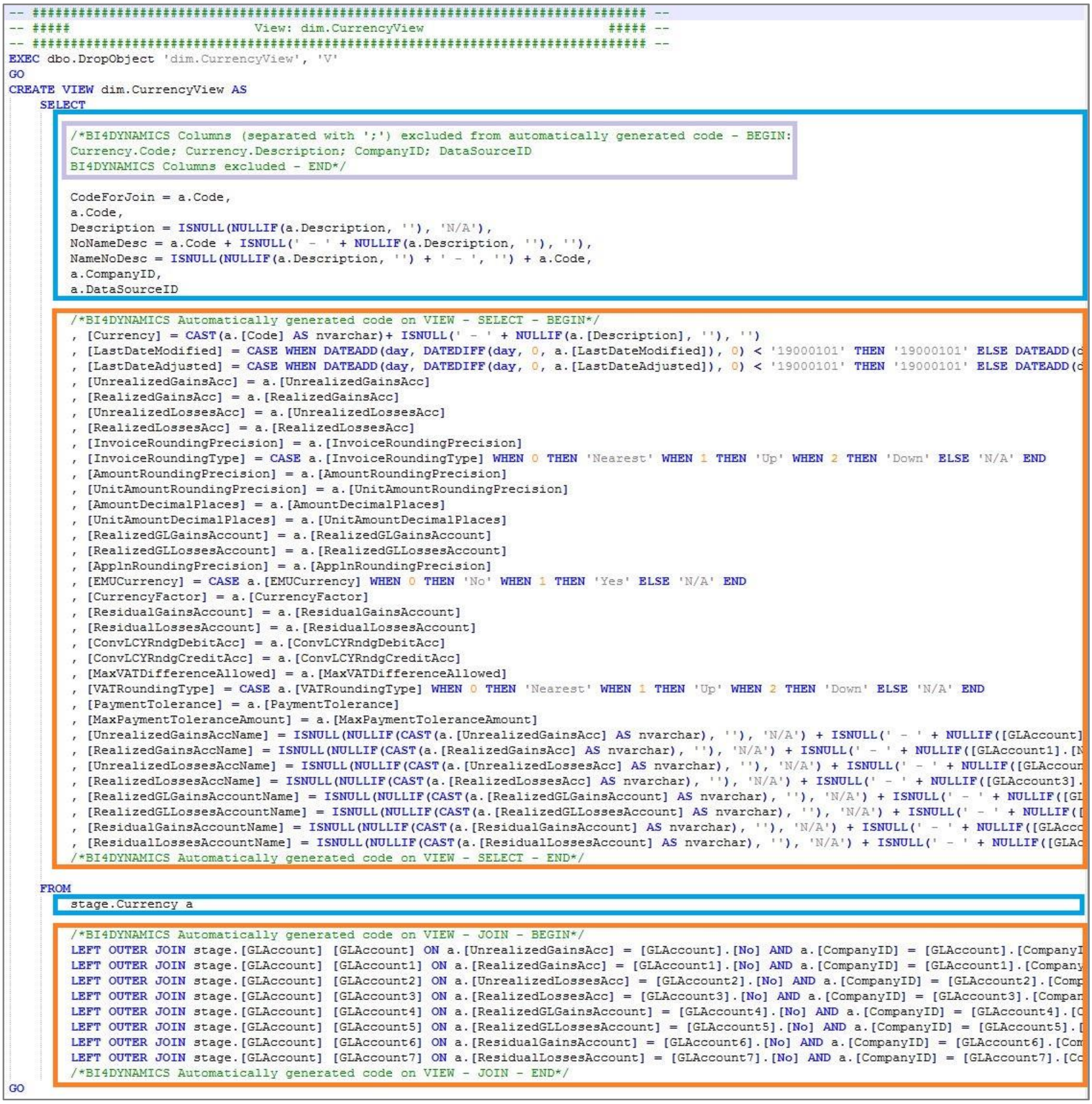
2.6.5 Scripts overview

All data warehouse scripts are divided into two parts:

- Standard code,
- Automatic (wizard) code.

That means that Wizard is inserting generated code into automatic parts, defined with placeholders at the beginning and end of such a part.
Further dimension customizations can also be made in standard part – that part is untouchable by Manage Dimensions.
Nevertheless automated code is marked a bit different in analytical file structure. Attributes have description **BI4-CUSTOM-ATTRIBUTE** and hierarchies are marked with **BI4-CUSTOM-HIERARCHY** description.

2.6.6 Migrate dimension properties from another instance



```sql
-- ################################################################################ --
-- #####                         View: dim.CurrencyView                     ##### --
-- ################################################################################ --
EXEC dbo.DropObject 'dim.CurrencyView', 'V'
GO
CREATE VIEW dim.CurrencyView AS
    SELECT
        /*BI4DYNAMICS Columns (separated with ';') excluded from automatically generated code - BEGIN:
        Currency.Code; Currency.Description; CompanyID; DataSourceID
        BI4DYNAMICS Columns excluded - END*/

        CodeForJoin = a.Code,
        a.Code,
        Description = ISNULL(NULLIF(a.Description, ''), 'N/A'),
        NoNameDesc = a.Code + ISNULL(' - ' + NULLIF(a.Description, ''), ''),
        NameNoDesc = ISNULL(NULLIF(a.Description, '') + ' - ', '') + a.Code,
        a.CompanyID,
        a.DataSourceID

        /*BI4DYNAMICS Automatically generated code on VIEW - SELECT - BEGIN*/
        , [Currency] = CAST(a.[Code] AS nvarchar)+ ISNULL(' - ' + NULLIF(a.[Description], ''), '')
        , [LastDateModified] = CASE WHEN DATEADD(day, DATEDIFF(day, 0, a.[LastDateModified]), 0) < '19000101' THEN '19000101' ELSE DATEADD(d
        , [LastDateAdjusted] = CASE WHEN DATEADD(day, DATEDIFF(day, 0, a.[LastDateAdjusted]), 0) < '19000101' THEN '19000101' ELSE DATEADD(d
        , [UnrealizedGainsAcc] = a.[UnrealizedGainsAcc]
        , [RealizedGainsAcc] = a.[RealizedGainsAcc]
        , [UnrealizedLossesAcc] = a.[UnrealizedLossesAcc]
        , [RealizedLossesAcc] = a.[RealizedLossesAcc]
        , [InvoiceRoundingPrecision] = a.[InvoiceRoundingPrecision]
        , [InvoiceRoundingType] = CASE a.[InvoiceRoundingType] WHEN 0 THEN 'Nearest' WHEN 1 THEN 'Up' WHEN 2 THEN 'Down' ELSE 'N/A' END
        , [AmountRoundingPrecision] = a.[AmountRoundingPrecision]
        , [UnitAmountRoundingPrecision] = a.[UnitAmountRoundingPrecision]
        , [AmountDecimalPlaces] = a.[AmountDecimalPlaces]
        , [UnitAmountDecimalPlaces] = a.[UnitAmountDecimalPlaces]
        , [RealizedGLGainsAccount] = a.[RealizedGLGainsAccount]
        , [RealizedGLLossesAccount] = a.[RealizedGLLossesAccount]
        , [ApplnRoundingPrecision] = a.[ApplnRoundingPrecision]
        , [EMUCurrency] = CASE a.[EMUCurrency] WHEN 0 THEN 'No' WHEN 1 THEN 'Yes' ELSE 'N/A' END
        , [CurrencyFactor] = a.[CurrencyFactor]
        , [ResidualGainsAccount] = a.[ResidualGainsAccount]
        , [ResidualLossesAccount] = a.[ResidualLossesAccount]
        , [ConvLCYRndgDebitAcc] = a.[ConvLCYRndgDebitAcc]
        , [ConvLCYRndgCreditAcc] = a.[ConvLCYRndgCreditAcc]
        , [MaxVATDifferenceAllowed] = a.[MaxVATDifferenceAllowed]
        , [VATRoundingType] = CASE a.[VATRoundingType] WHEN 0 THEN 'Nearest' WHEN 1 THEN 'Up' WHEN 2 THEN 'Down' ELSE 'N/A' END
        , [PaymentTolerance] = a.[PaymentTolerance]
        , [MaxPaymentToleranceAmount] = a.[MaxPaymentToleranceAmount]
        , [UnrealizedGainsAccName] = ISNULL(NULLIF(CAST(a.[UnrealizedGainsAcc] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GLAccount]
        , [RealizedGainsAccName] = ISNULL(NULLIF(CAST(a.[RealizedGainsAcc] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GLAccount1].[N
        , [UnrealizedLossesAccName] = ISNULL(NULLIF(CAST(a.[UnrealizedLossesAcc] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GLAccoun
        , [RealizedLossesAccName] = ISNULL(NULLIF(CAST(a.[RealizedLossesAcc] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GLAccount3].
        , [RealizedGLGainsAccountName] = ISNULL(NULLIF(CAST(a.[RealizedGLGainsAccount] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GL
        , [RealizedGLLossesAccountName] = ISNULL(NULLIF(CAST(a.[RealizedGLLossesAccount] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([
        , [ResidualGainsAccountName] = ISNULL(NULLIF(CAST(a.[ResidualGainsAccount] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GLAcco
        , [ResidualLossesAccountName] = ISNULL(NULLIF(CAST(a.[ResidualLossesAccount] AS nvarchar), ''), 'N/A') + ISNULL(' - ' + NULLIF([GLAc
        /*BI4DYNAMICS Automatically generated code on VIEW - SELECT - END*/

    FROM
        stage.Currency a

        /*BI4DYNAMICS Automatically generated code on VIEW - JOIN - BEGIN*/
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount] ON a.[UnrealizedGainsAcc] = [GLAccount].[No] AND a.[CompanyID] = [GLAccount].[CompanyI
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount1] ON a.[RealizedGainsAcc] = [GLAccount1].[No] AND a.[CompanyID] = [GLAccount1].[Company
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount2] ON a.[UnrealizedLossesAcc] = [GLAccount2].[No] AND a.[CompanyID] = [GLAccount2].[Comp
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount3] ON a.[RealizedLossesAcc] = [GLAccount3].[No] AND a.[CompanyID] = [GLAccount3].[Compan
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount4] ON a.[RealizedGLGainsAccount] = [GLAccount4].[No] AND a.[CompanyID] = [GLAccount4].[C
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount5] ON a.[RealizedGLLossesAccount] = [GLAccount5].[No] AND a.[CompanyID] = [GLAccount5].[
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount6] ON a.[ResidualGainsAccount] = [GLAccount6].[No] AND a.[CompanyID] = [GLAccount6].[Com
        LEFT OUTER JOIN stage.[GLAccount] [GLAccount7] ON a.[ResidualLossesAccount] = [GLAccount7].[No] AND a.[CompanyID] = [GLAccount7].[Co
        /*BI4DYNAMICS Automatically generated code on VIEW - JOIN - END*/
GO
```

Dimension configuration files from subfolder Dimension in folder 6-Wizard can be coped to any BI4Dynamics instance with same BI4Dynamics version and same (or at least similar) NAV structure.

All information about hidden attributes, custom hierarchies, status and daily update from .bi4dim file will append to related dimensions available in Manage Dimensions menu when opening.

3 HOW TO SECTION

This section has examples of how Wizard functionality can be applied in practice.

3.1.1 Manage errors

Most errors will appear due to duplication of dimension names. Check names before generating scripts.

1 Fact have disappeared from cube

If deploy finished with error that it has deleted staging part of BI4Dynamics where NAV metadata are. User can see the cubes, but not facts that have been created.

Solution:

1. Go to Development tab and bring NAV metadata back. NAV metadata is checked.
2. Uncheck table NAV metadata
3. Check it again
4. Click update button

Facts will appear again.

3.1.2 Manage dimension

1 Change action from existing dimension to existing field

> Example 1: Changing action in manage dimension for field User (related table T2000000002) Wizard will suggest to connect fields from source tables to existing dimension. User can change suggested action from Existing dimension to Existing fields when NAV metadata do not have required information for Dimension creations. Dimension will be built from all codes that are in source table (for example T17) and not from related table (T2000000002)

3.1.3 Manage measures

1 Examples of custom MDX

User has to replace **Net Change** with measure that Function will be calculated for.

Function	Formula
BALANCE	SUM(PERIODSTODATE([Date].[Date YMD].[(All)], [Date].[Date YMD]), [Measures].[Net Change])