bi4dynamics
Transforming your data into knowledge

BI4Dynamics Cube Wizard White paper

May 2015

This document is based on BI4Dynamics NAV version 5.0.2. Built 870.

Content

1 CUBE WIZARD FEATURES 3
1.1 NAV metadata 4
1.2 BI templates 4
1.3 Add fact to any standard BI4Dynamics cube 5
1.4 Work simultaneously with standard BI4Dynamics, Wizard and Microsoft SQL server tools 5
1.5 Language features 5
1.6 Distribution of Wizard solution file 6
2 PREPARATION #1: MANAGE NAV METADATA 7
2.1 Introduction 7
2.2 Install Microsoft Dynamics NAV objects 7
2.3 Change object number 8
2.4 Run report to create NAV metadata 9
2.5 How to import table object when customer doesn't have free table 10
3 PREPARATION #2: CREATE INSTANCE 11
3.1 Create BI4Dynamics instance 11
3.2 Import NAV metadata table to BI4Dynamics 11
4 CREATE CUBE WITH CUBE WIZARD 13
4.1 Step 1: Manage cube 13
4.2 Step 2: Select source table 14
4.3 Step 3: Select fields 15
4.4 Step 4: Manage dimensions 16
4.5 Step 5: Manage measures 18
4.6 Save solution and generate scripts 19
5 MANAGE WIZARD GENERATED CUBE 21
5.1 Edit existing fact from Wizard Cube 21
5.2 Add new fact to Wizard Cube 21
5.3 Copy cube from another instance 21
6 MANAGE EXISTING STANDARD BI4DYNAMICS CUBE 22
6.1 Add new fact to existing standard BI4Dynamics cube 22
6.2 Edit existing fact 22
6.3 Delete Wizard generated fact that has been added to standard cube 23
7 RUNNING V5 AND PREVIOUS VERSION SIMULTANEOUSLY 24
8 HOW TO SECTION 26
8.1 Manage errors 26
8.2 Manage dimension 26
8.3 Manage measures 26

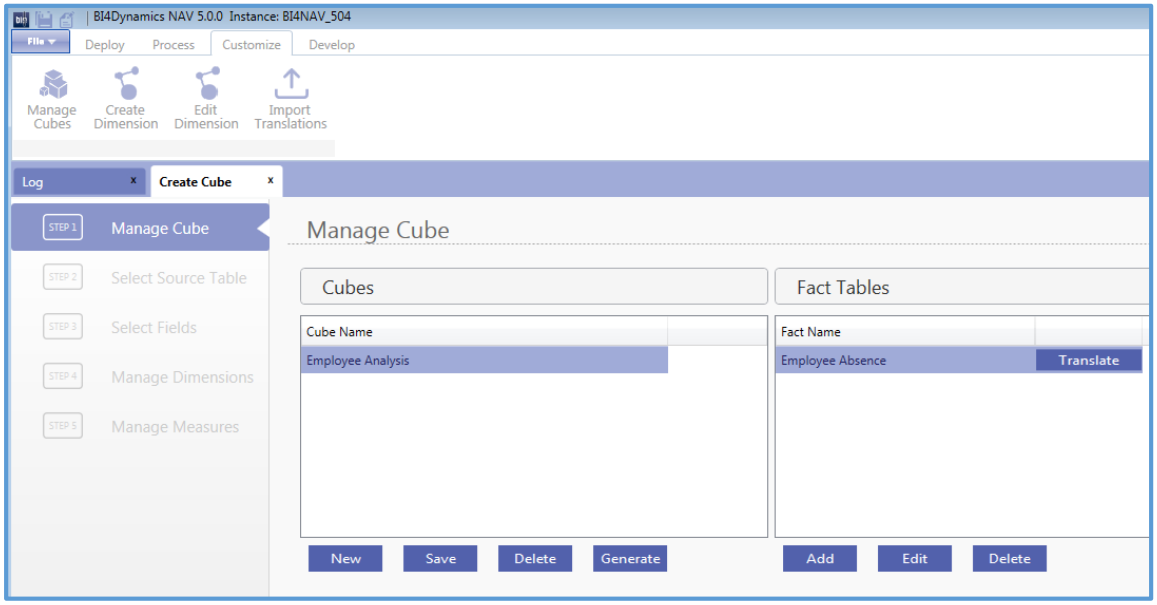
Total pages: **26**

1 CUBE WIZARD FEATURES

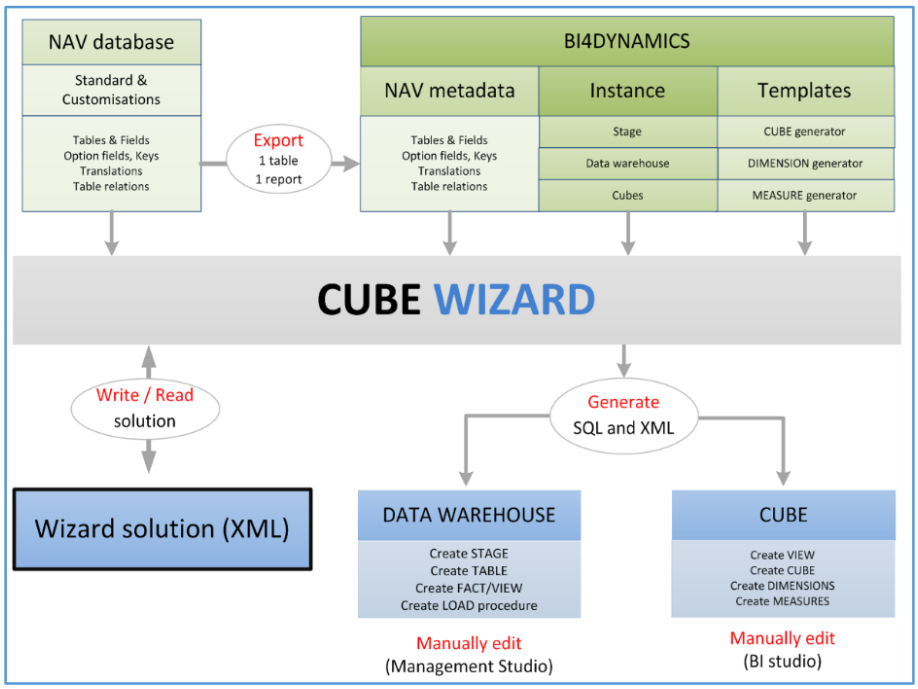
BI4Dynamics NAV Cube Wizard easily creates custom Cubes in 5 simple steps:

1. Manage Cube
2. Select Source Table
3. Select Fields
4. Manage Dimensions
5. Manage Measures

It does not require any SQL or other technical knowledge.



Cube Wizard steps



Logical schema of BI4Dynamics Cube Wizard

1.1 NAV metadata

1 Logical meaning

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

2 Physical meaning

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 Cube 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.

1.2 BI templates

BI templates are building block of Cube Wizard. They are predefined functionality that is used by Cube Wizard. A visible part of BI templates is seen through user interface where user can select different options and choose different set-up actions. Invisible part of BI templates are SQL and XML script, hidden in Cube Wizard that are generated at the end of cube creation process. Cube Wizard is currently supporting these templates:

1.2.1 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.

1.2.2 Dimensions

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

  BI4Dynamics supports 99% of all table relations in NAV.

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

1.2.3 Dimension attributes

- All fields are added as dimension attributes to DW and cube. Flow fields and flow filters are not included.
- Created are only those attributes that have values in NAV; Option, Boolean Field Types are always shown because it is not possible to find if they have value or not.

1.2.4 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;

1.2.5 Measures

- Simple data warehouse based measures: SUM, MIN, MAX, COUNT, DISTINCT COUNT
- Measure groups:
    - YTD (year-to-date),
    - MTD (month-to-date),
    - WTD (month-to-date),
    - YOY (year over year),
    - LAST PERIODS (one measure for each of last 12 months)
    - POP (period over period)
- All custom MDX is supported

1.3 Add fact to any standard BI4Dynamics 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;

1.4 Work simultaneously with standard BI4Dynamics, Wizard and Microsoft SQL server tools

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

Solution created by BIDS (Business intelligence Development Studio) can be saved to BI4Dynamics folder, deployed and processed just like cubes created by Wizard. Following process is supported:

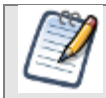
1. Create cube with standard deploy process or with cube wizard
2. Open and modify cube with BIDS
3. Save BIDS solution to folder (5-Develpment)
4. Deploy and process instance with BI4Dynamics

1.5 Language features

- Every language available in NAV database is also available in cubes created by Cube 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

1.6 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.

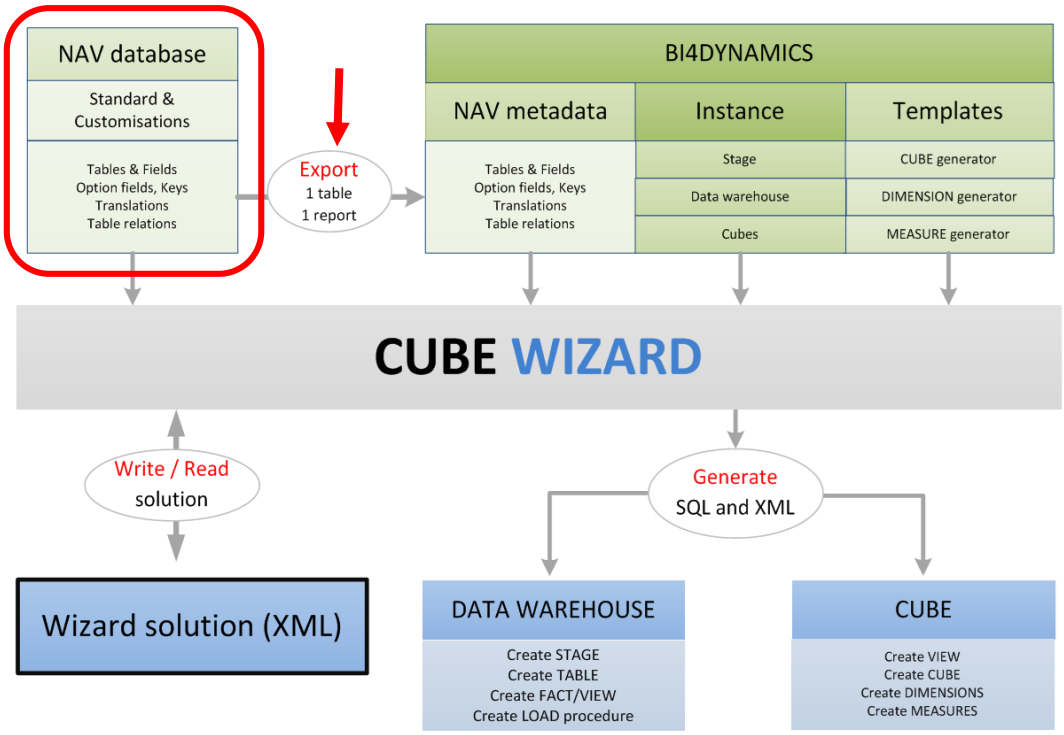
	Wizard may work also when NAV structures (NAV metadata) in source and target installation are not the same, but must have same NAV languages!

2 PREPARATION #1: MANAGE NAV METADATA

This section shows how to create NAV metadata for NAV 2009. The same process is used to create NAV metadata for other versions of NAV.

2.1 Introduction

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



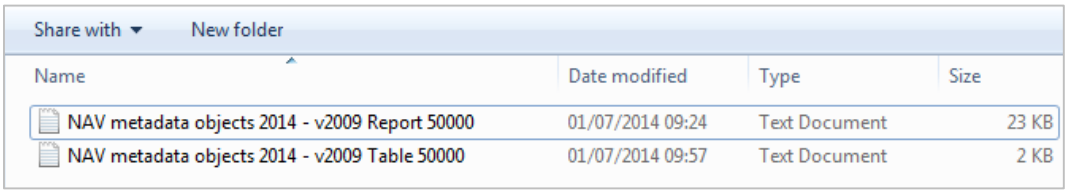
PREREQUISITE NAV OBJECTS

Customer needs to have free NAV objects: **1 table, 1 report;** it's up to customer to provide these free objects. BI4Dynamics do not provide NAV license with these objects. Object files may be different for different NAV versions and are provided by BI4Dynmics.

2.2 Install Microsoft Dynamics NAV objects

There are 2 NAV object needed to create NAV metadata:

- 1 Table: used to store NAV metadata
- 1 Report: used to read NAV metadata and writes them to table



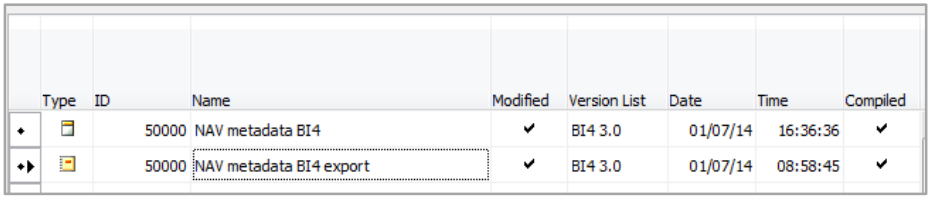
Share with ▾ New folder

Name	Date modified	Type	Size
NAV metadata objects 2014 - v2009 Report 50000	01/07/2014 09:24	Text Document	23 KB
NAV metadata objects 2014 - v2009 Table 50000	01/07/2014 09:57	Text Document	2 KB

Picture shows 2 txt files containing 1 report and 1 table for NAV 2009.

2.2.1 Import Objects in Object Designer

If customer has empty numbers for 2 NAV objects than these can be imported in NAV Object Designer (Tools\Object designer or Shift-F12)



Type	ID	Name	Modified	Version List	Date	Time	Compiled
	50000	NAV metadata BI4	✔	BI4 3.0	01/07/14	16:36:36	✔
	50000	NAV metadata BI4 export	✔	BI4 3.0	01/07/14	08:58:45	✔

Picture shows two NAV objects needed to manage NAV metadata.

2.3 Change object number

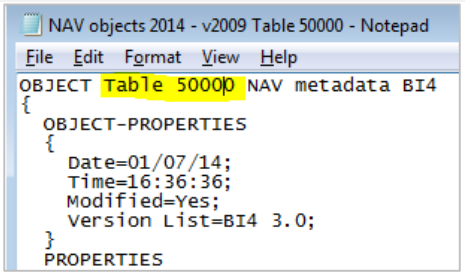
Object can have any number. In this example we will change the numbers of objects:

- Table: from 50000 to 50009
- Report: from 50000 to 50050

2.3.1 Changing the Number for NAV table

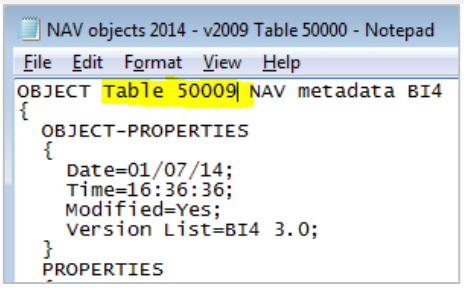
In this example there is an empty table with number 50009. Use Notepad to open txt object (name is not important) with NAV object for Table. In first row of txt file change the number 50000 to new number 50009:

Before



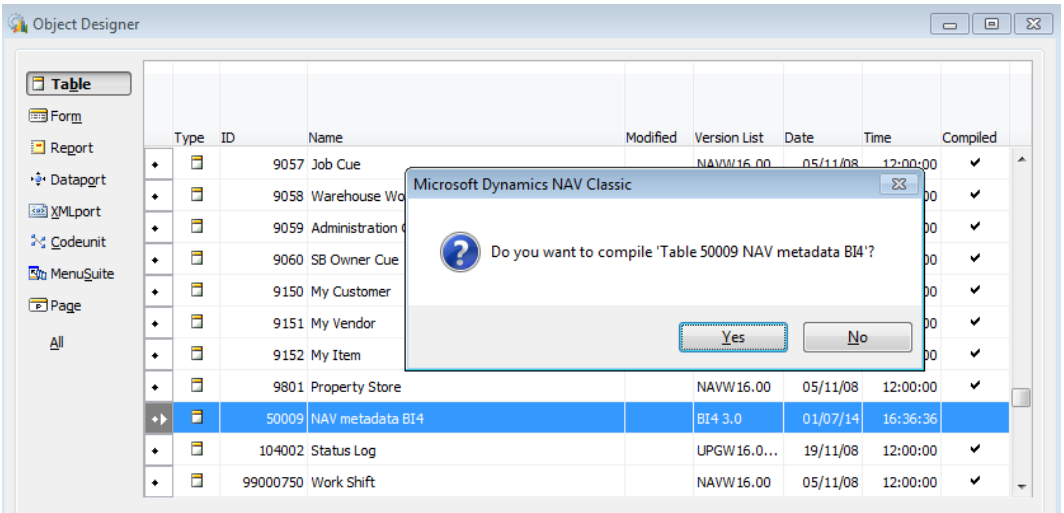
```
OBJECT Table 50000 NAV metadata BI4
{
  OBJECT-PROPERTIES
  {
    Date=01/07/14;
    Time=16:36:36;
    Modified=Yes;
    Version List=BI4 3.0;
  }
  PROPERTIES
```

After



```
OBJECT Table 50009 NAV metadata BI4
{
  OBJECT-PROPERTIES
  {
    Date=01/07/14;
    Time=16:36:36;
    Modified=Yes;
    Version List=BI4 3.0;
  }
  PROPERTIES
```

Save the file,

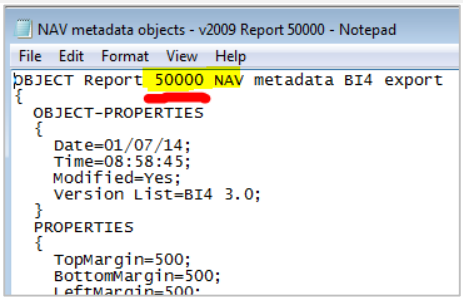


Open Object Designer (Shift-F12), import file (File \ Import) and compile it (select table and press F11).

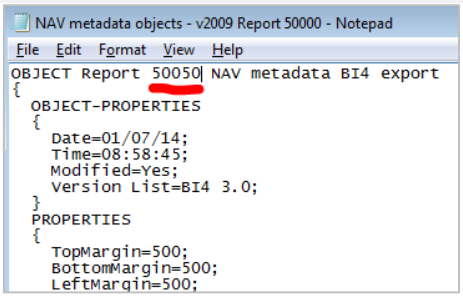
2.3.2 Changing the Number for NAV report

Use Notepad to open report object and change number:

Original report number



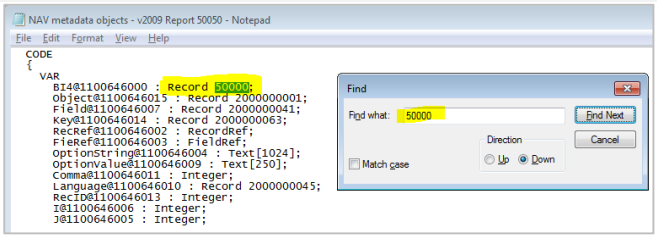
Changed report number



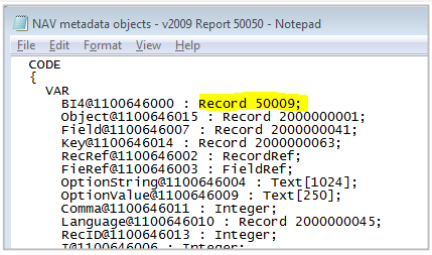
2.3.3 Change number of NAV table in NAV report

In the same report search for number of table 50000 and change it to new number (50009). Number has to be changed only once!

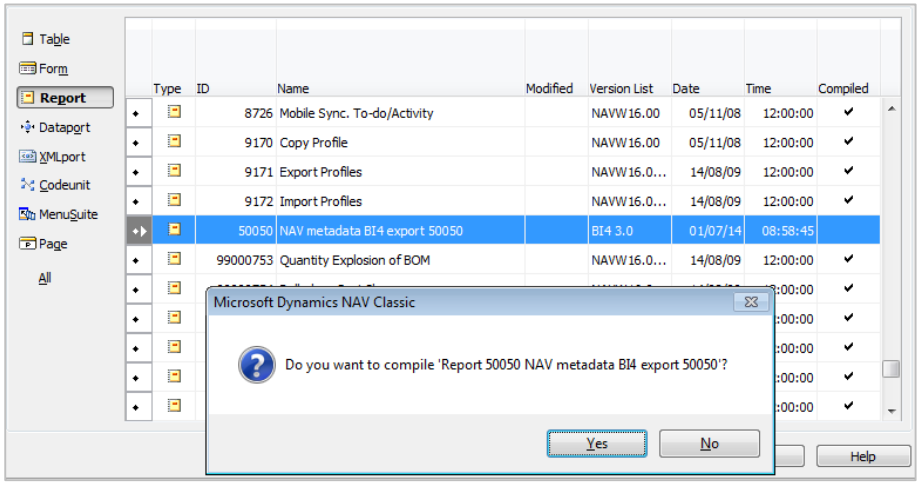
Before:



After:



We save this txt file and import it to NAV in Object designer.



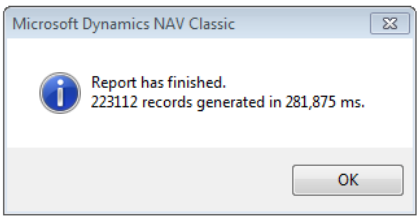
Since report has been imported as txt objects it has to be compiled (select report and press F11).

2.4 Run report to create NAV metadata

2.4.1 Run report

Run report 50050 (no requested filter fields, no window dialog) by selecting report and clicking “Preview” button. Report will finish generating data in few minutes (1’ to 5’). Time depends on hardware and number of languages used in NAV database.

At the end report will finish processing by information window telling Number of records and time needed to process:



2.4.2 Check NAV metadata

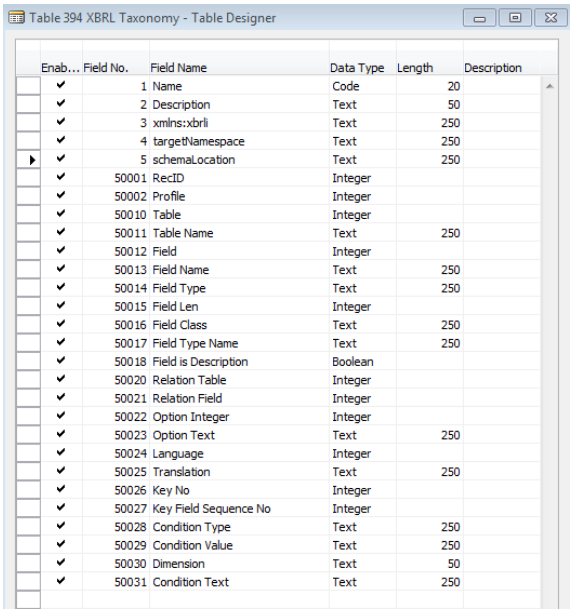
Run table 50009 and check number of records. If there are records jobs is done successfully. NAV installation with 1 language has around 65k records (English language code 1033), installation with 4 languages (English + 3 additional) around 125k records in this table.

2.5 How to import table object when customer doesn't have free table

2.5.1 Find one licensed table not used in NAV

NAV has 900+ tables. Very probably not all tables are used (differs from customer to customer). Find one table in license range that is not used.

Example: Table 394 "XBRL Taxonomy"



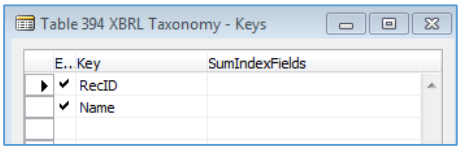
Enab...	Field No.	Field Name	Data Type	Length	Description
✔	1	Name	Code	20	
✔	2	Description	Text	50	
✔	3	xmlns:xbrli	Text	250	
✔	4	targetNamespace	Text	250	
✔	5	schemaLocation	Text	250	
✔	50001	RecID	Integer		
✔	50002	Profile	Integer		
✔	50010	Table	Integer		
✔	50011	Table Name	Text	250	
✔	50012	Field	Integer		
✔	50013	Field Name	Text	250	
✔	50014	Field Type	Text	250	
✔	50015	Field Len	Integer		
✔	50016	Field Class	Text	250	
✔	50017	Field Type Name	Text	250	
✔	50018	Field is Description	Boolean		
✔	50020	Relation Table	Integer		
✔	50021	Relation Field	Integer		
✔	50022	Option Integer	Integer		
✔	50023	Option Text	Text	250	
✔	50024	Language	Integer		
✔	50025	Translation	Text	250	
✔	50026	Key No	Integer		
✔	50027	Key Field Sequence No	Integer		
✔	50028	Condition Type	Text	250	
✔	50029	Condition Value	Text	250	
✔	50030	Dimension	Text	50	
✔	50031	Condition Text	Text	250	

2.5.2 Copy fields from "NAV metadata BI4" table

All fields in NAV metadata BI4 table are above 50000. You can copy fields from this table to "XBRL Taxonomy". If you can't copy this ask your NAV partner to do it with partner license or write fields manually into table.

2.5.3 Add primary key

A primary key "Record ID" must be added to table.



E..	Key	SumIndexFields
✔	RecID	
✔	Name	

2.5.4 Change NAV table number in report

Check instruction in previous chapter and change the NAV table number in report from 50000 (default) to 394 (our example).

NAV part is done.

2.5.5 Change the staging name of table

Instance has to be created first and changing the staging name of table has to be done next.
Please follow the instruction in next chapter where this is described.

3 PREPARATION #2: CREATE INSTANCE

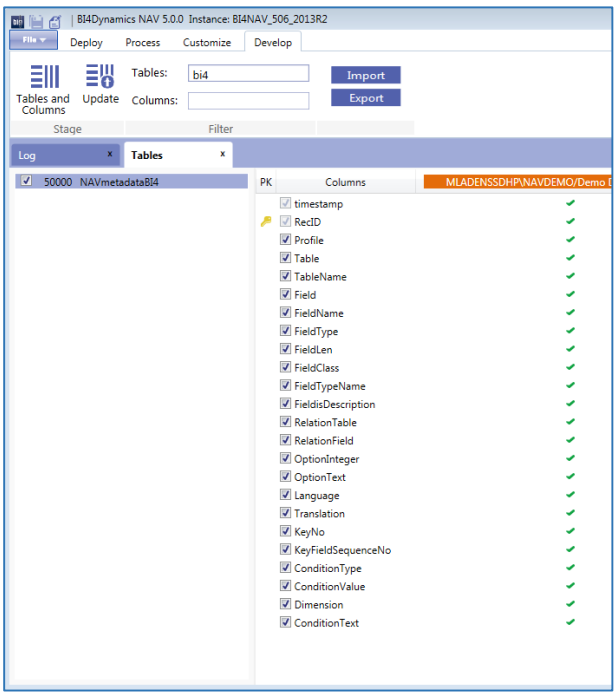
3.1 Create BI4Dynamics instance

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

3.2 Import NAV metadata table to BI4Dynamics

NAV metadata are created in NAV table and are in NAV part of SQL database. This table has to be brought to BI part SQL database of BI4Dynamics. Process has two options:

- Import new NAV metadata table (T50000)
- Import re-used NAV metadata table (example T394)



3.2.1 Import new NAV metadata table to BI (staging area of SQL server)

This process will bring NAV metadata table from NAV to BI (= staging area of SQL instance):

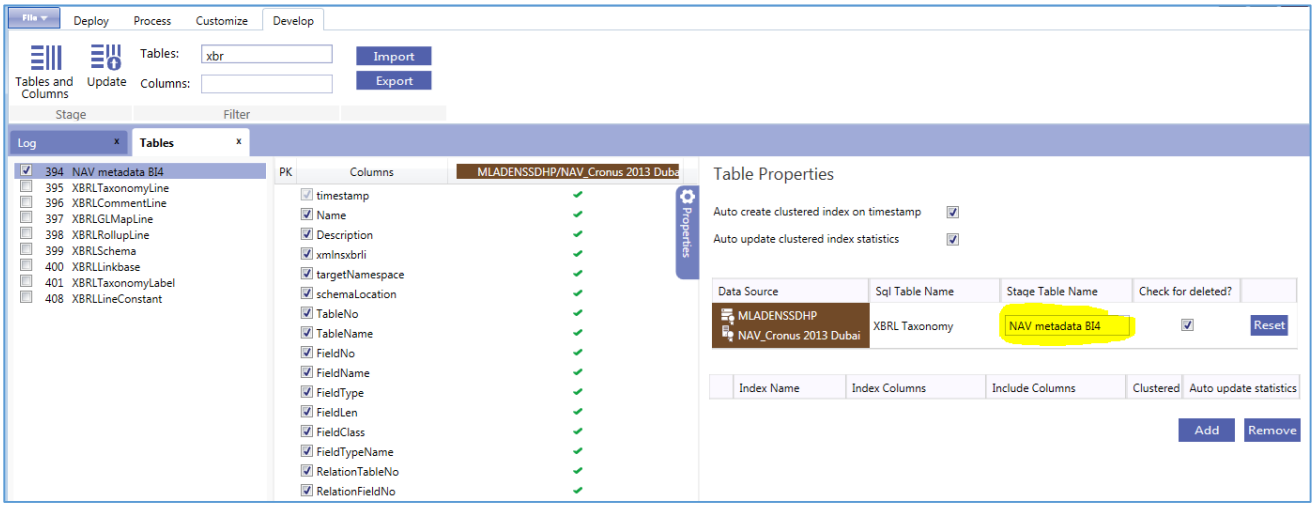
1. **Go to "Develop" tab**
2. **Click on "Tables and Columns" icon**
3. **Enter "BI4" into "Tables" field to filter a table that contains Meta data (by default NAVmetadataBI4)**
4. **Check table in "Tables" tab**
5. **Click "Update" icon; it takes few seconds to bring NAV metadata to staging area of SQL instance**

This process has to be done only once, when you start using Cube Wizard.

3.2.2 How to import table object when customer doesn't have free table

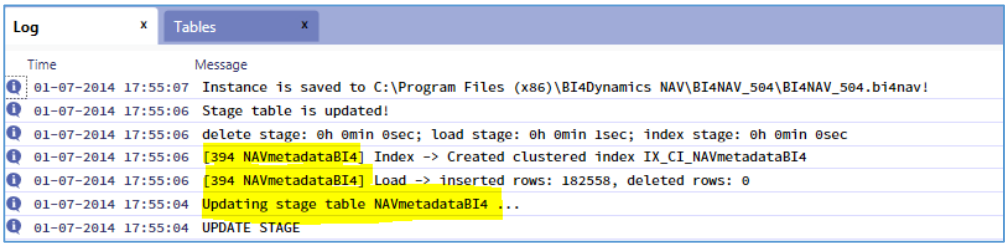
In this option we import re-used NAV table "XBRL Taxonomy" Table 394 to SQL database for BI. Step 3 is different from standard procedure as we are looking table with different name

1. **Go to "Develop" tab**
2. **Click on "Tables and Columns" icon**
3. **Changed: Enter "XBR" into "Tables" field to filter a table that contains Meta data**
4. **Check table in "Tables" tab**
5. **New step - Rename: On the right part of screen rename Stage table name to "NAV metadata BI4". Name must be exactly the same.**
6. **Click "Update" icon; it takes few seconds to bring NAV metadata to staging area of SQL instance**



7. **Check log**

Go to "Log" tab in check latest lines. Log should have lines similar to these:



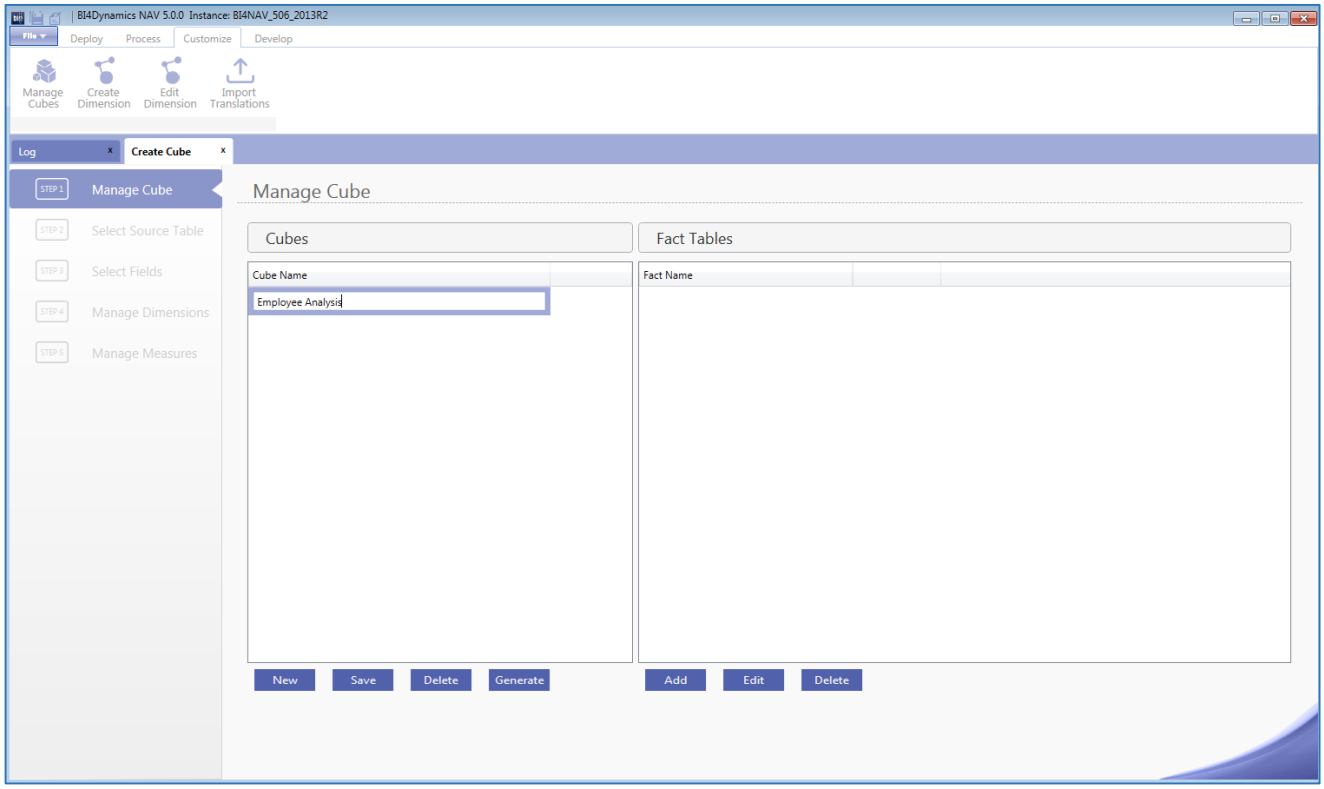
4 CREATE CUBE WITH CUBE WIZARD

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.

4.1 Step 1: Manage cube

1 **Go to "Customize" tab**
2 **Click on "Manage Cube"**
3 **Click on "New"**
4 **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.



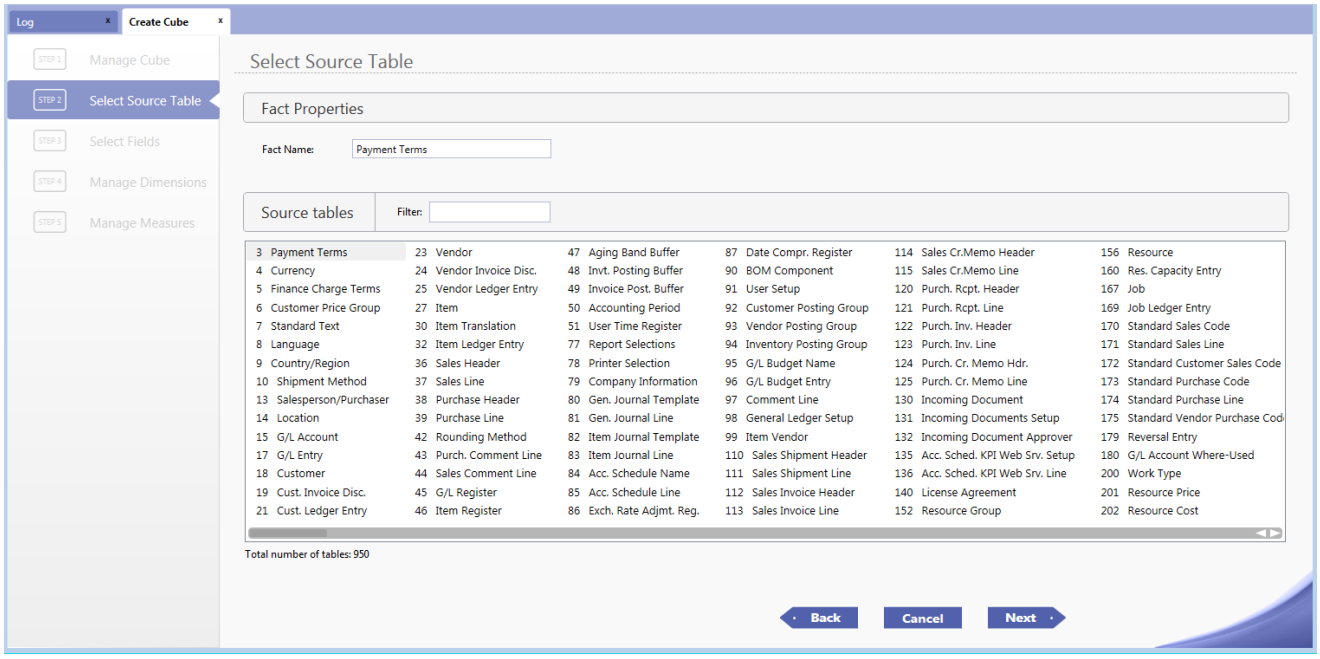
Example of cube name: Employee Analysis

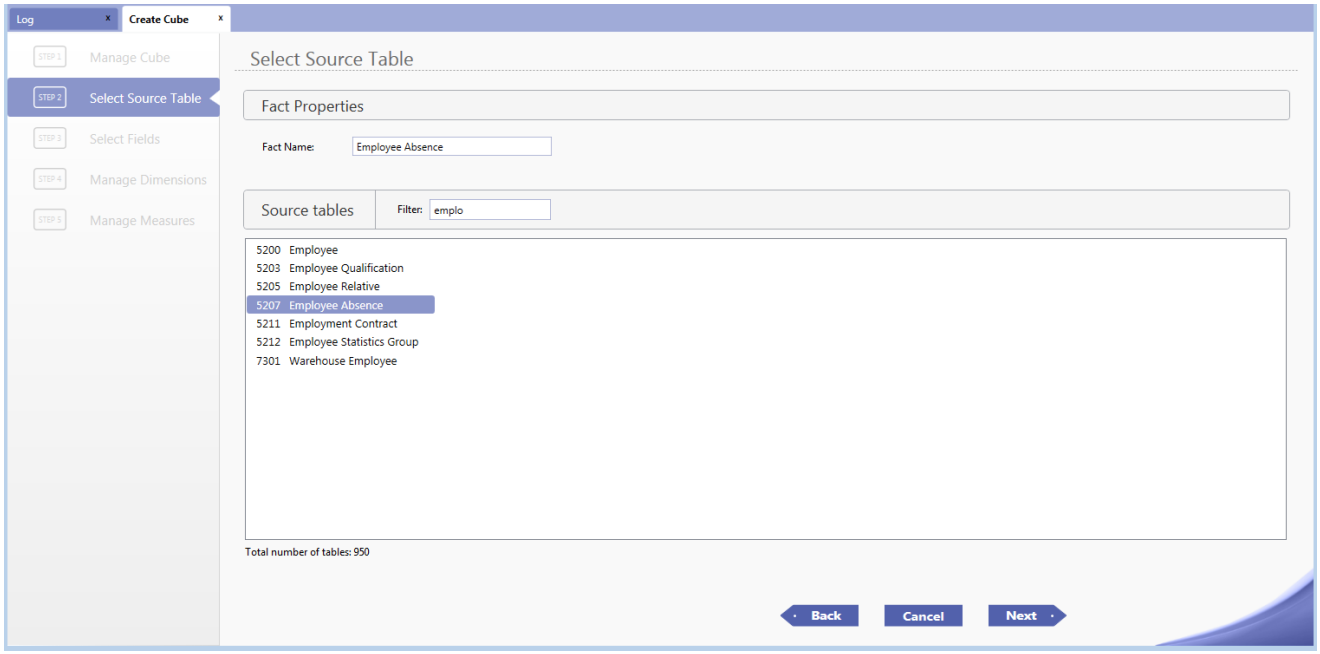
5 **Click "Add"**

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

4.2 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.





1 **Use filter to find source table**

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

2 **Click “Next” to move to next step**

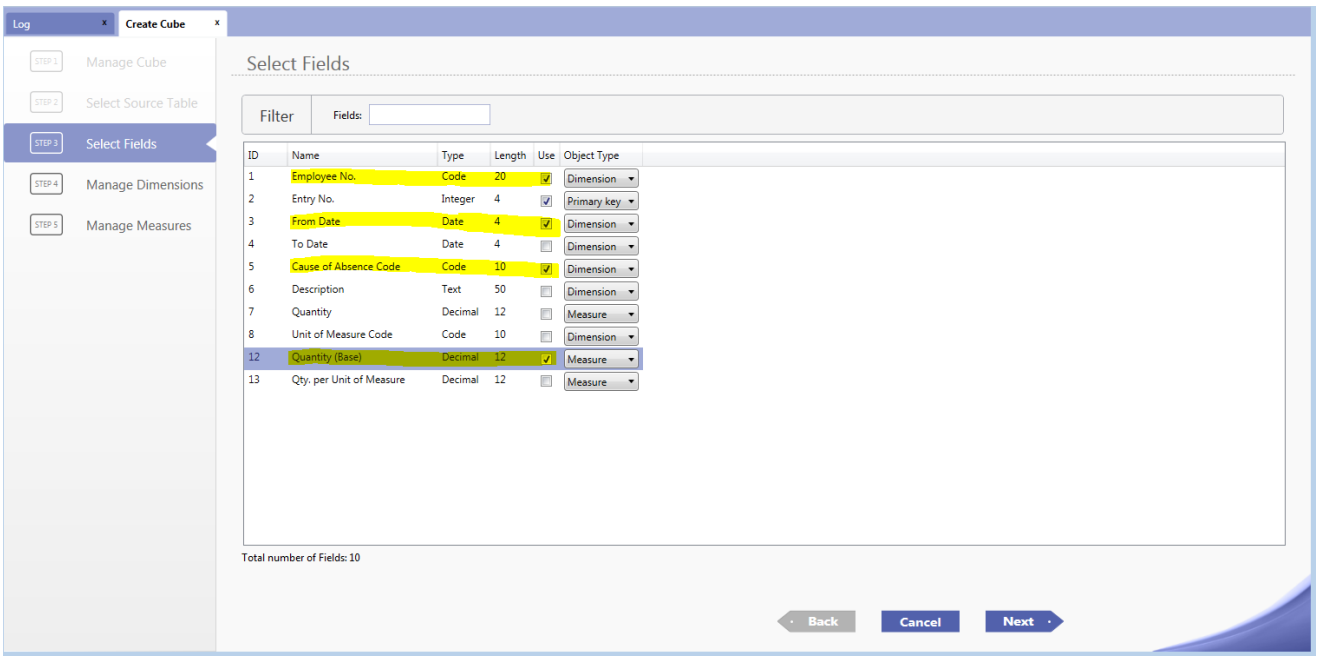
4.3 Step 3: Select fields

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

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



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.

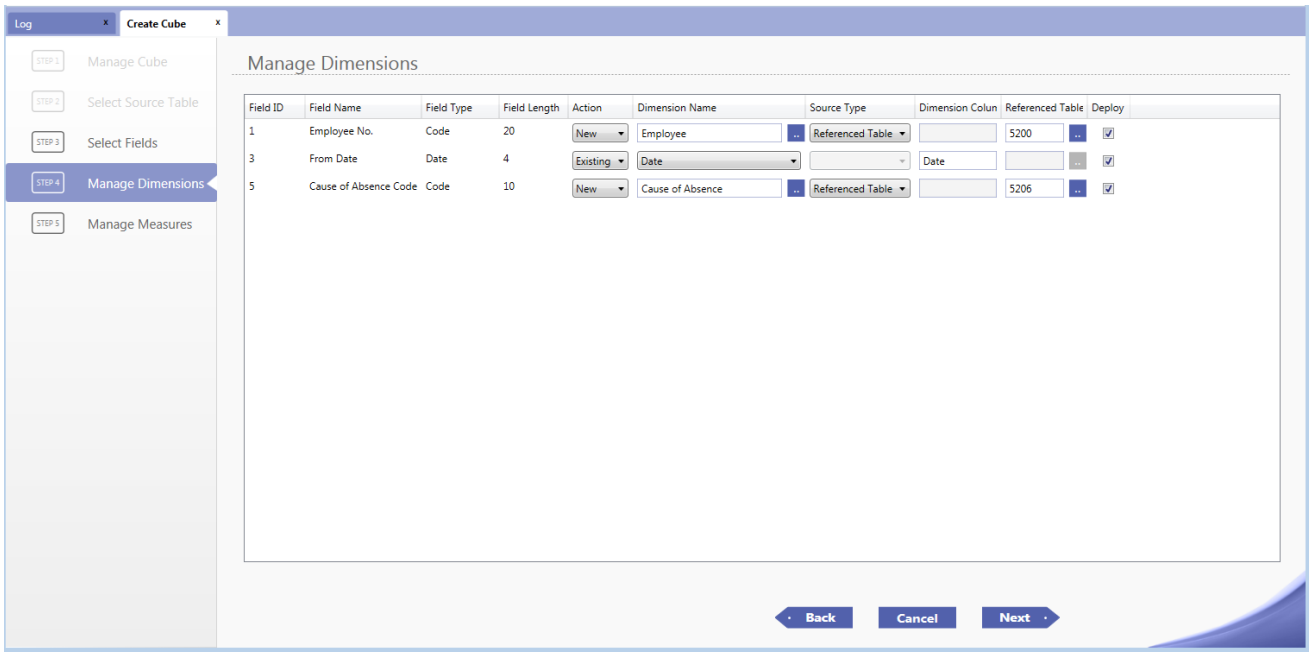
4.4 Step 4: Manage dimensions

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:

4.4.1 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



4.4.2 Dimension name

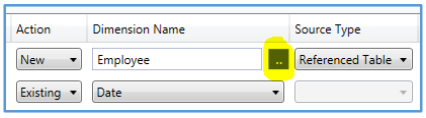
1 Select new Dimension

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.

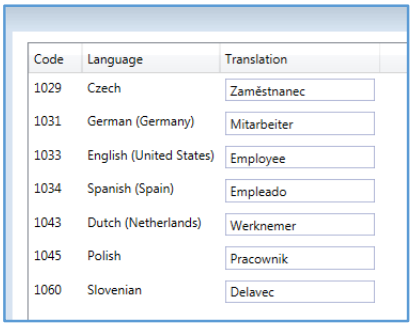
This field can be changed if new dimension is created and we would like to change the name.

2 Change translations

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



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

3 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.

4.4.3 Source Type

Source of dimension can be:

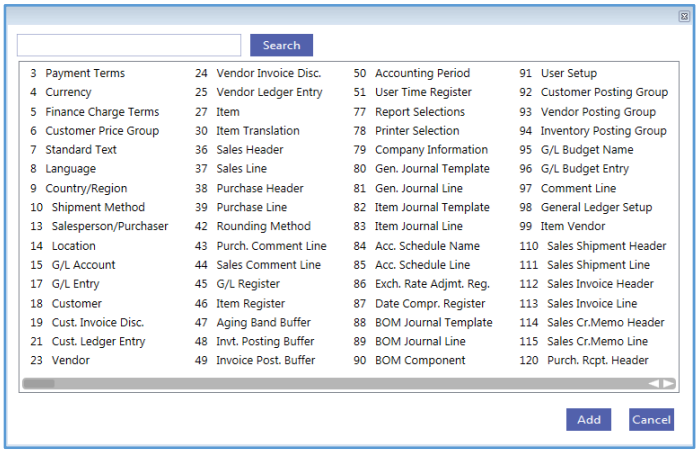
- Reference table
- Current field

4.4.4 Dimension column

1. Select measures and dimensions you want to include
2. Next

4.4.5 Reference table ID

This is a number of NAV table that will create a new dimension. We can select any table by clicking blue section at the end of field:



4.4.6 Reference field

If new dimension has been selected usually this field is filed automatically with primary key of selected table. If this field is empty that fields that presents a primary field has to be selected manually.

Note: name has to be selected from selected NAV table from BI database, not from NAV.

4.4.7 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.

- 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

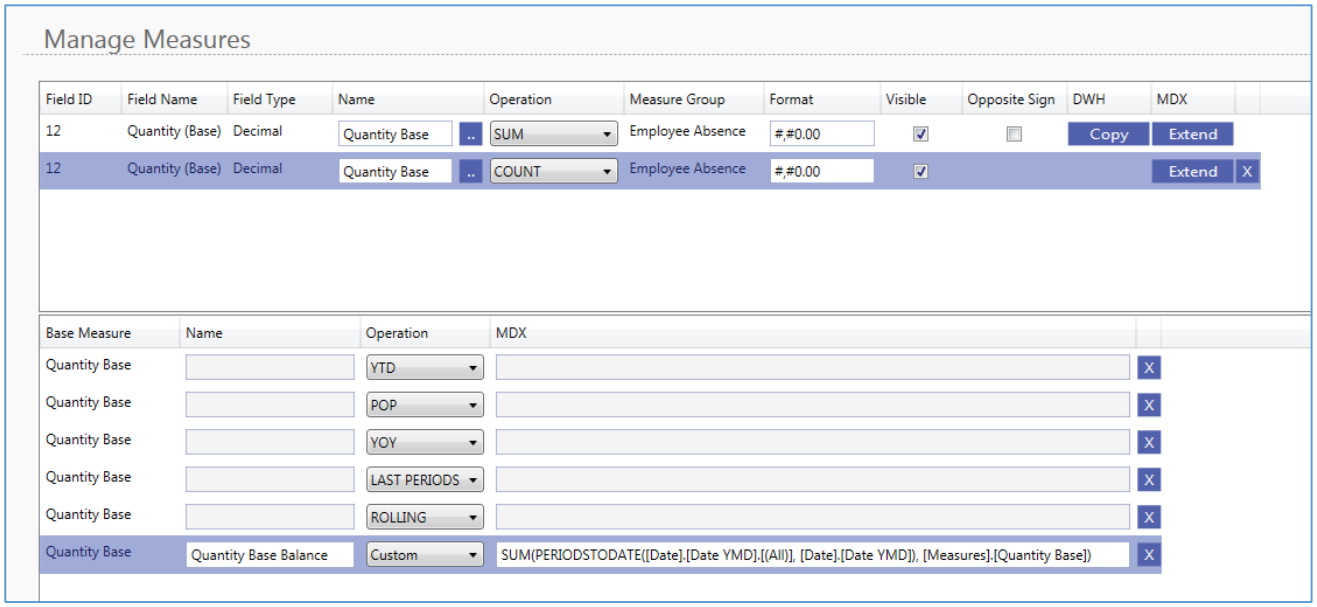
Both fields can be connected to existing dimension Customer

4.4.8 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.

4.5 Step 5: Manage measures



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

This is description of fields in this step:

4.5.1 Name

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

4.5.2 Operation

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

4.5.3 Measure group

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

4.5.4 Format

Propose is most common decimal format.

4.5.5 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.

4.5.6 Opposite sign

Check this field if you want to change a sign.

4.5.7 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.

4.5.8 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.

1 **Add predefined measure groups:**

- YTD add 5 measures
- LAST PERIODS adds 12 measures

2 **Add custom MDX scripts**

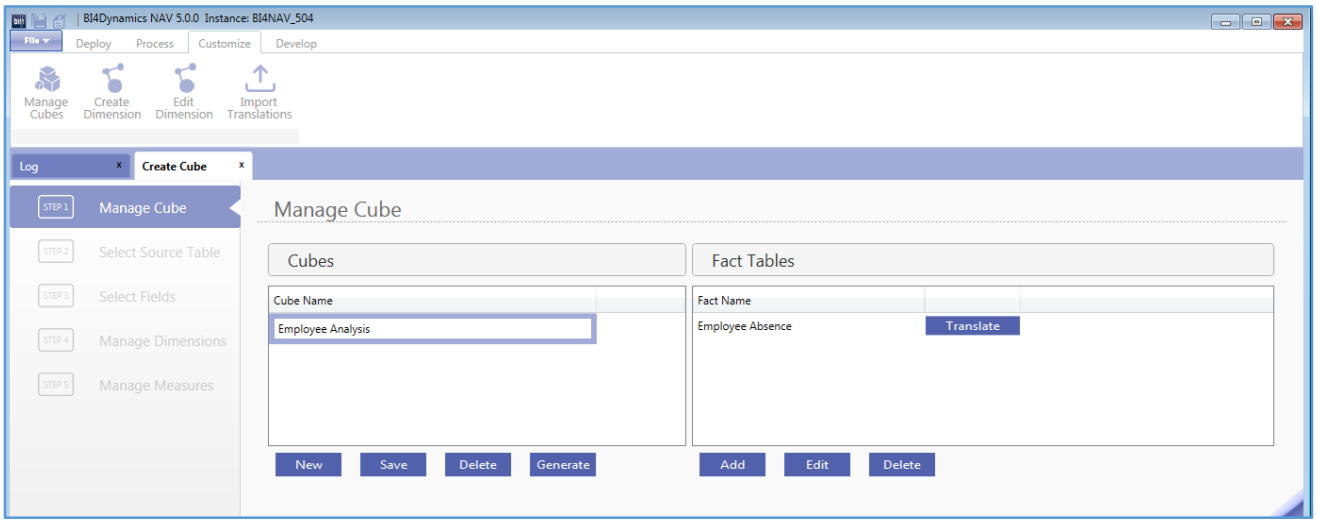
By selecting option Custom any MDX script can be added. Wizard will not check if MDX is correct. This has to be done (if needed) outside wizard in SQL tools.

4.5.9 Next

By selecting "Next" changes are written in memory.

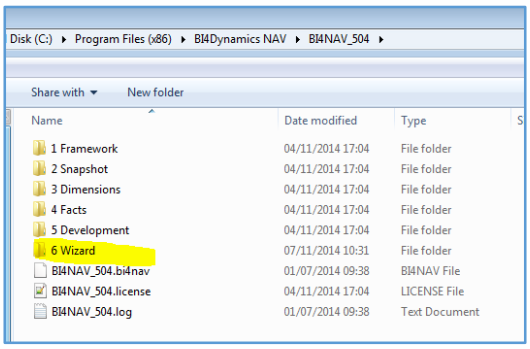
4.6 Save solution and generate scripts

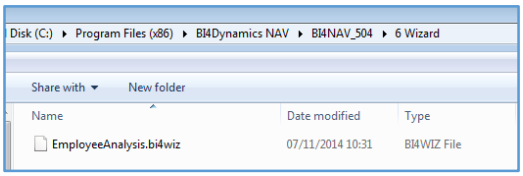
Wizard is at step 1, where we can save wizard solution and generate scripts.



4.6.1 Save

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





4.6.2 Generate

Cube 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.

4.6.3 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.

5 MANAGE WIZARD GENERATED CUBE

5.1 Edit existing fact from Wizard Cube

We can edit cube or fact by clicking Edit. Do not forget to save changes and generate scripts

5.2 Add new fact to Wizard Cube

New facts can be added to existing cubes;

5.2.1 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

5.2.2 Manage measures

1 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.

2 Calculate any measure

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

5.3 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 Cube wizard all files that are in folder “6-Wizard” are presented as cubes.

6 MANAGE EXISTING STANDARD BI4DYNAMICS CUBE

6.1 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.

6.2 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.

6.2.1 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

6.2.2 Error message

Measure group with name “Fact_name_entry” already exists

6.2.3 Correct process

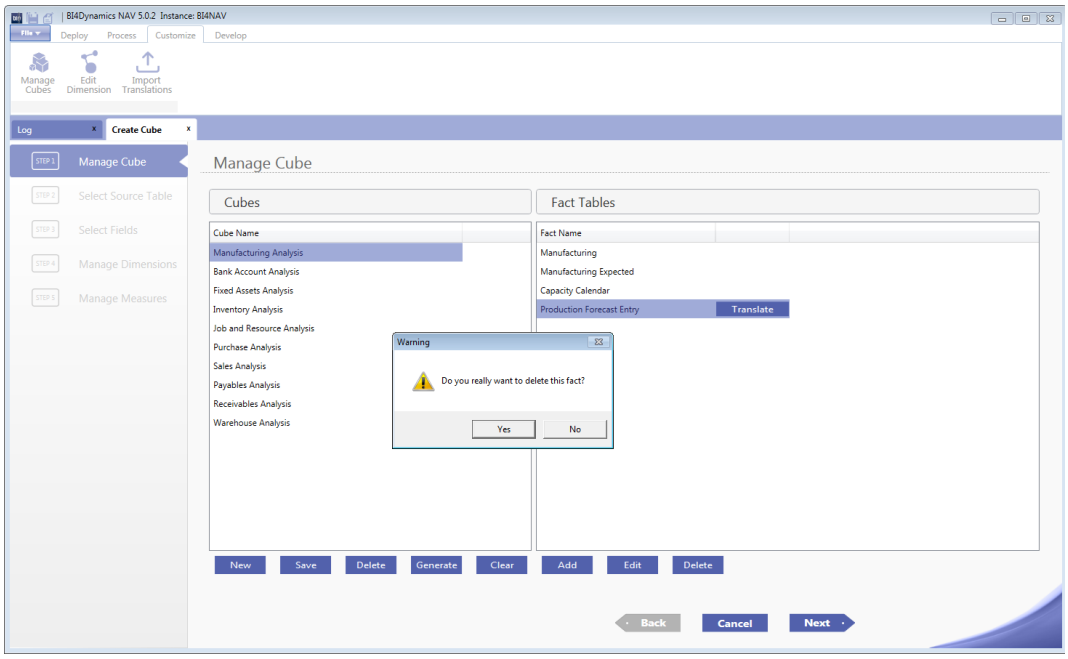
1. …
2. Save cube, generate scripts, deploy & process cube
3. **DELETE FOLDER 5 WITH FACT NAME**
4. **DEPLOY CUBE**
5. Edit fact
6. Save cube
7. Generate scripts (no error)

6.2.4 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.

6.3 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.



6.3.1 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

6.3.2 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

6.3.3 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

Need to do only one step.

7 RUNNING V5 AND PREVIOUS VERSION SIMULTANEOUSLY

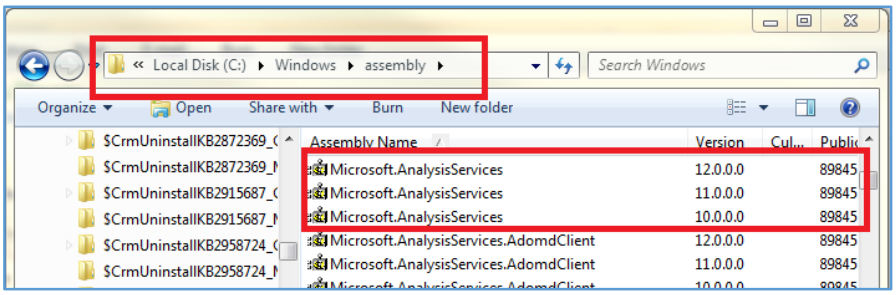
BI4Dynamics v5 is running on latest version of Microsoft AMO / SMO assemblies. They are automatically installed by the installation wizard. This can lead to .dll conflicts when installing this version in parallel with older BI4Dynamics versions. Application configuration file on older versions needs to be updated so that older version is properly started with the correct version of SMO/AMO assemblies. A zip package with properly updated configuration files for older versions is included in release along with release notes and these installation instructions.

To run all versions side by side follow these instructions:

Don't uninstall v3/v4 via installer!

1 **Make sure AMO/SMO version 11.0.0.0 is present at system**

BI4Dynamics installer adds these files.



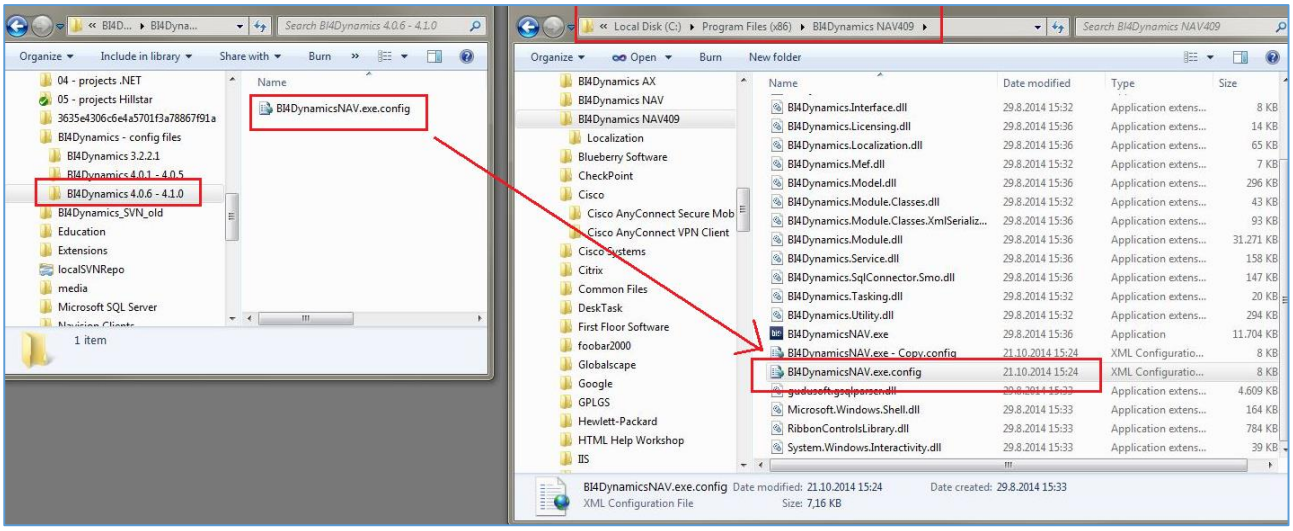
2 **Backup "BI4DynamicsNAV" installation folder**

This backup may be used to reverse the process in case of any malfunction.

3 **Rename existing BI4Dynamics folder and add version postfix**

BI4Dynamics NAV -> BI4Dynamics NAV409

4 **Replace old config file with a new one**

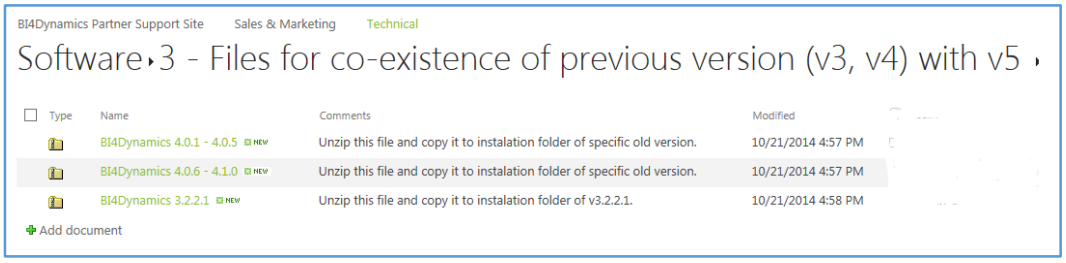


Choose "copy & replace" after "Drag & Drop" action, do not rename new file.

5 Install (upgrade) BI4Dynamics 5

Use **upgrade** option in installation wizard of version 5 and follow installer instructions

Files are available on partner portal:



Ask your partner to get these files.

8 HOW TO SECTION

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

8.1 Manage errors

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

8.1.1 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 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.

8.2 Manage dimension

8.2.1 Change action from existing dimension to existing field

Example 1: Changing action in manage dimension for field User (related table T2000000002) Cube 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)

8.3 Manage measures

8.3.1 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])