

# GetSetUniverse 3.5.26 Documentation

<b>Author</b>	<b>Andre Luetzkendorf</b>
<b>Date</b>	<b>07.02.2010</b>
<b>Contact</b>	<a href="http://www.luetzkendorf.eu">www.luetzkendorf.eu</a> <a href="mailto:info@luetzkendorf.eu">info@luetzkendorf.eu</a>
© Changes or other modifications of the documentation will need an authorisation of the author.	

1	Summary.....	3
2	Installation.....	4
3	Reports .....	6
3.1	sheet info_ <universe name> .....	6
3.2	sheet parameter_ <universe name> .....	8
3.3	sheet objects_ <universe name> .....	9
3.4	sheet dependents_ <universe name> .....	11
3.5	sheet contexts_ <universe name> .....	12
3.6	sheet hier_ <universe name> .....	13
3.7	sheet joins_ <universe name> .....	14
3.8	sheet incomb_ <universe name> .....	15
4	Options .....	16
5	Load Universe .....	17
5.1	Process .....	17
5.1.1	Run Load Universe .....	17
5.1.2	Log on as Designer .....	17
5.1.3	Select universe .....	17
5.1.4	Result .....	17
6	Write Universe.....	18
6.1	Process .....	18
6.1.1	Preconditions.....	18
6.1.2	Changing the object properties.....	18
6.1.3	Start Write Universe.....	18
6.1.4	Log of changes .....	19
6.1.5	Check the results .....	19
7	Versioning Universe .....	20
7.1	Process .....	20
7.1.1	Auswahl Optionen .....	20
7.1.2	Start Versioning Universe .....	20
7.1.3	Log on as Designer .....	20
7.1.4	Select the Universe .....	20
7.2	Result .....	20
8	Error handling .....	21
8.1	Error during reading the universe data.....	21
8.2	Additional Errors .....	21

# 1 Summary

*GetSetUniverse* supports the Business Objects Universe designer by

- transparent reporting of all universe objects
- impact/ lineage analysis (presentation of dependencies) (@functions, incompatibilities)
- listing of used parameter within the universe
- versioning of universes

*GetSetUniverse* is executable for version SAP BO XI.

## 2 Installation

*GetSetUniverse* will be delivered within an excel file. For a successful execution the following environment will be needed:

- installation of Business Objects clients
- installation of Microsoft Excel (with VBA)
- installation of the following libraries:
  - ▣ MSO.DLL (Microsoft Office 11.0 Object Library)
  - ▣ stdole2.tlb (OLE Automation)
  - ▣ VBE6.DLL (Visual Basic For Applications)

The excel file *GetSetUniverse.xls* includes a macro. The security of excel has to allow the execution of the macro.

With opening the excel file an additional menu item *GetSetUniverse* will appear. (as the older code was written german the german meaning is listed in brackets)

The following sub items are defined:

Menu item	Meaning
Load universe (Lade Universum)	Universe objects will be loaded directly into the excel file <i>GetSetUniverse</i> .
Write universe (Schreibe Universum)	The loaded object data can be modified and written back into the universe.  Changeable data are the properties name, description and select of an universe object
Versioning Universe >> xls (Versioniere Universum >> xls)	A new excel document will be generated. It includes all exportable data of the universe. With usage of a diff tool (like winmerge) further versions of the universe can be compared.
Options (Optionen)	Options <ul style="list-style-type: none"> <li>■ manage the level of detail which will be stored after running &lt;Versioning Universe&gt;</li> <li>■ define the language of menu items and headers of output</li> </ul>
Info	Info

Additional useful macros can be used under <tools> <macro><macros ...>.

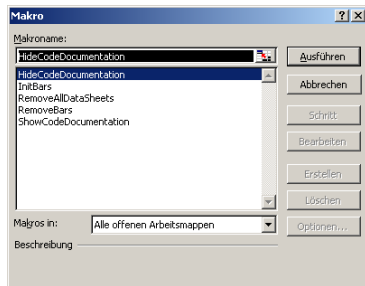


Abbildung 1 Verfügbare Makros

Macro	Meaning
ShowCodeDocumentation	shows internal sheets, which describes the structure of internal used objects (the description is in german)
HideCodeDocumentation	hides internal sheets
InitBars	Generates menu item <i>GetSetUniverse</i> and related sub items
RemoveBars	Removes menu item <i>GetSetUniverse</i>
RemoveAllDataSheets	Removes all data sheets whose contains imported universe data

### 3 Reports

Below the generated reports as result of the executed menu items

- *Load universe*
- *Versioning universe*

will be described.

#### 3.1 sheet info\_<universe name>

The sheet shows the properties of the universe.

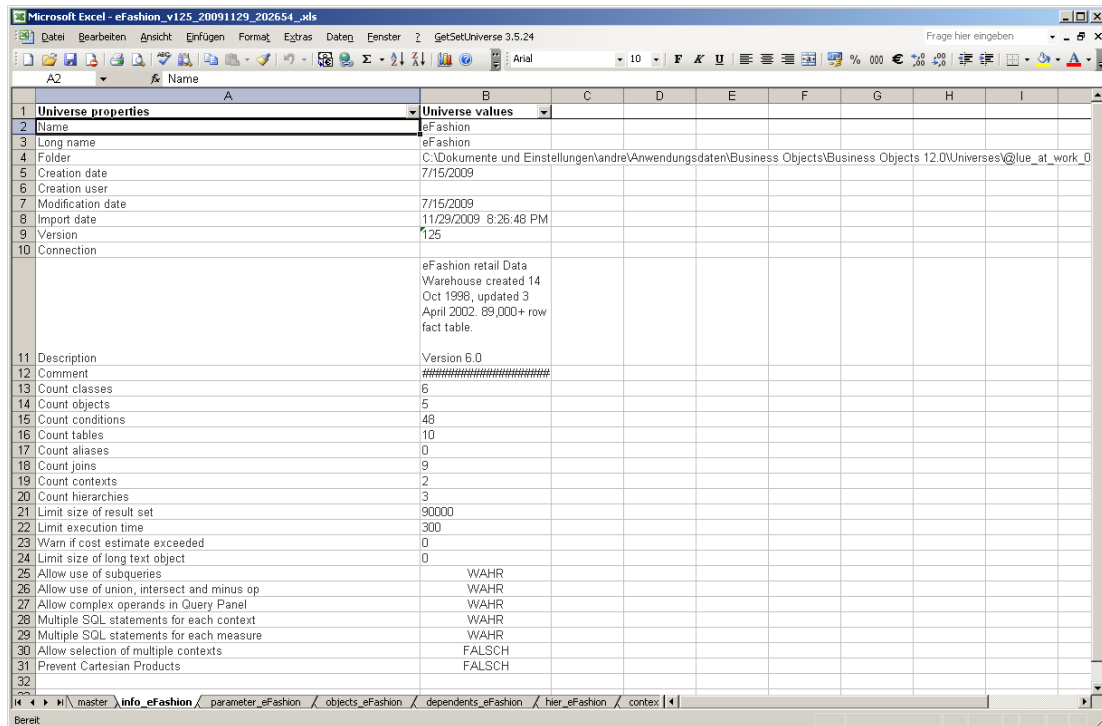
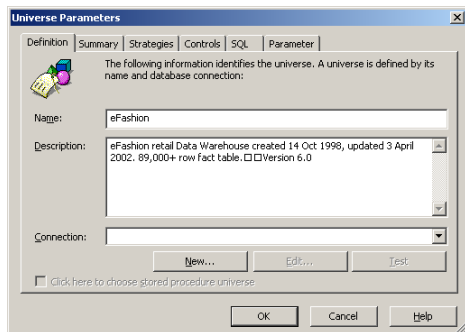


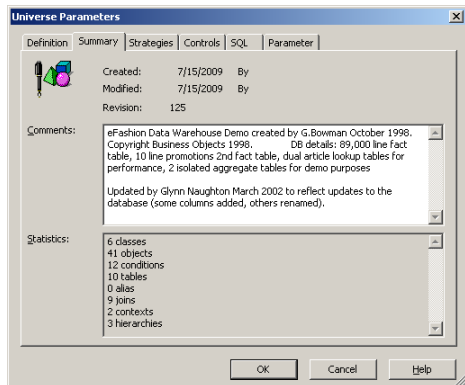
Abbildung 2 Mappe info\_eFashion

Properties	Meaning
Name	Name of the universe (stored file name)
Long name	Long name of the universe (see Pic 3 Universe Parameters sheet Definition)
Folder	Locale folder from which the universe will be imported
Creation date	Creation date of the universe (see Pic 4 )
Creation user	Creation user of the universe
Modification date	Modification date of the universe (see Pic 4 )
Import date	Import date from CMS
Version	version of the universe (see Pic 4 )
Connection	Connection name of the universe (see Pic 4 )
Description	Description of the universe (see Pic 3 Universe Parameters sheet Definition)
Comment	Internal comment of the universe (see Pic 4 )
Counts	Count classes, objects, conditiond, etc. (see Pic 4 )

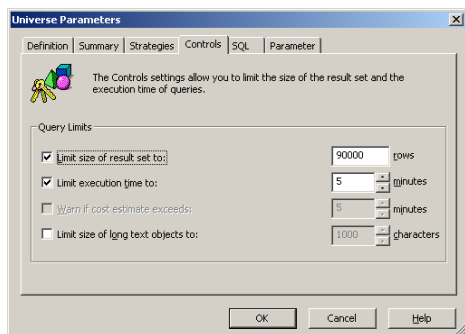
Properties	Meaning
Restriction	Restrictions (see Pic 5 )
SQL options	Sql options (see Pic 6 )



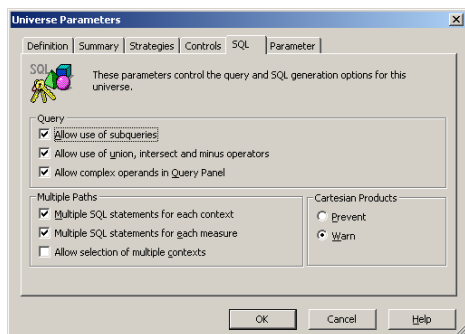
Pic 3 Universe Parameters sheet Definition



Pic 4 Universe Parameters sheet Summary



Pic 5 Universe Parameters sheet Controls



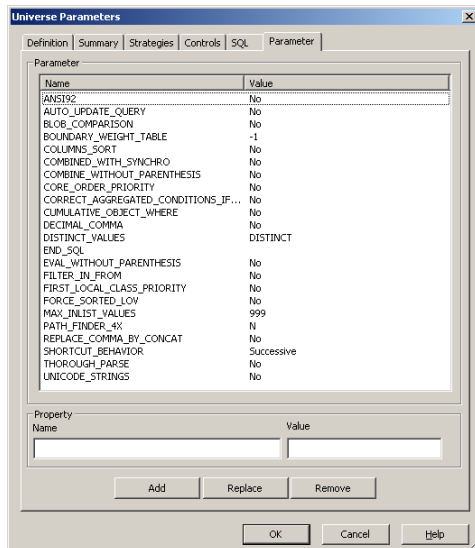
Pic 6 Universe Parameters sheet SQL

### 3.2 sheet parameter\_ <universe name>

The sheet shows the database parameter of the universe.

Parameter	Value
ANSIS2	No
AUTO_UPDATE_QUERY	No
BLOB_COMPARISON	No
BOUNDARY_WEIGHT_TABLE	-1
COLUMNS_SORT	No
COMBINED_WITH_SYNCRO	No
COMBINE_WITHOUT_PARENTHESIS	No
CORE_ORDER_PRIORITY	No
CORRECT_AGGREGATED_CONDITIONS_IF_DRILL	No
CUMULATIVE_OBJECT_WHERE	No
DECIMAL_COMMA	No
DISTINCT_VALUES	DISTINCT
END_SQL	
EVAL_WITHOUT_PARENTHESIS	No
FILTER_IN_FROM	No
FIRST_LOCAL_CLASS_PRIORITY	No
FORCE_SORTED_LOV	No
MAX_INLIST_VALUES	999
PATH_FINDER_4X	N
REPLACE_COMMA_BY_CONCAT	No
SHORTCUT_BEHAVIOR	Successive
THOROUGH_PARSE	No
UNICODE_STRINGS	No

Pic 7 sheet parameter\_eFashion

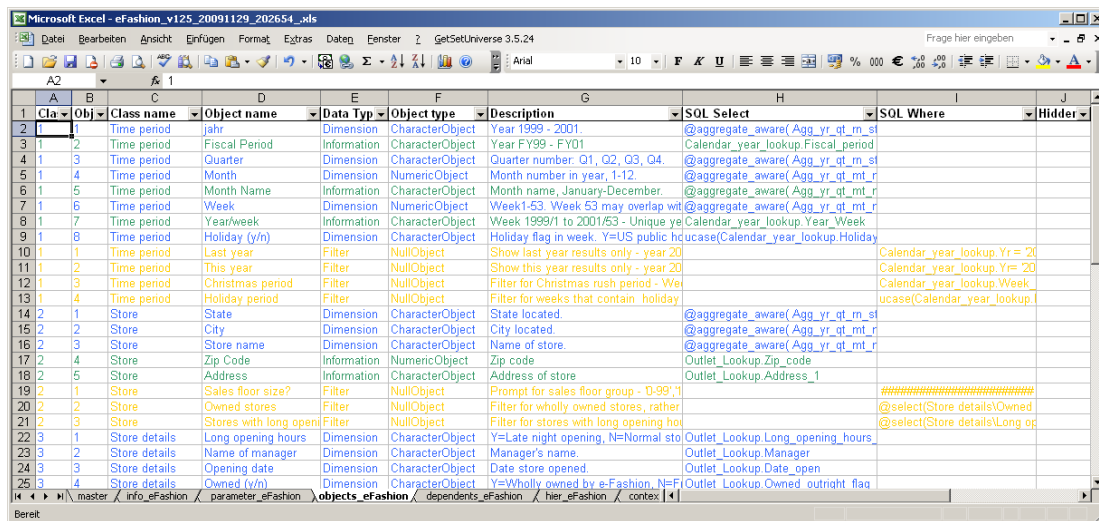


Pic 8 Universe Parameters sheet Parameter



### 3.3 sheet objects\_<universe name>

All object data of the universe will be stored within the sheet objects\_<universe name>.



Pic 9 sheet objects\_eFashion

The meaning of all columns will be described below. The values of the columns

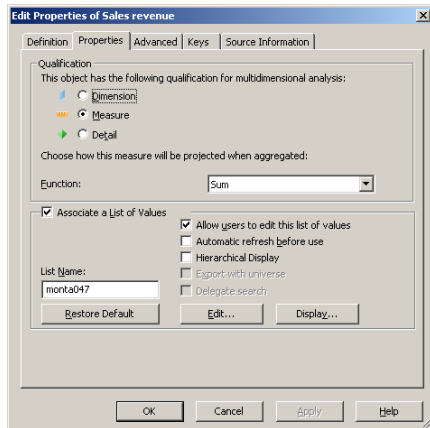
- Description
- SQL Select
- SQL Where

are editable. The user can modify these object data and writing back the changes after executing the program *Write unverse (Schreibe Universum)*. All other data are not editable.

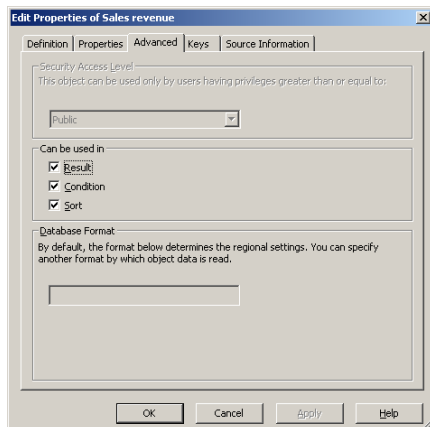
**Avoid a re-sorting of the rows. After then an export of the data back into the universe can fail.**

column	meaning	Editable
Class ID	Id of object class	
Object ID	Id of object	
Class name	name of object class	
Object name	name of object	
Data type	Data type of object	
Object type	Typ of object <span style="color:blue">Dimension</span> dimension objects will be marked blue <span style="color:green">Information</span> detail objects will be marked green <span style="color:red">Kennzahl</span> measure objects will be marked red <span style="color:yellow">Filter</span> condition objects will be marked yellow	
Description	Description of the object	X
SQL Select	Select-statement of the object	X
SQL Where	Where-condition of the object	X
Hidden	Hidden objects will be marked with "x".	
The following object data will only be reported by using the program Versioning universe.		
Aggregat function	see Pic 10 properties of an object	
Associate a List of Values	see Pic 10 properties of an object	
List name	see Pic 10 properties of an object	
Allow users to edit this list of values	see Pic 10 properties of an object	
Automatic refresh before use	see Pic 10 properties of an object	
Hierarchical Display	see Pic 10 properties of an object	

column	meaning	Editable
Export with universe	see Pic 10 properties of an object	
Security Access Level	see Pic 11 Advanced properties of an object	
can be used in: Result	see Pic 11 Advanced properties of an object	
can be used in: Condition	see Pic 11 Advanced properties of an object	
can be used in: Sort	see Pic 11 Advanced properties of an object	



Pic 10 properties of an object

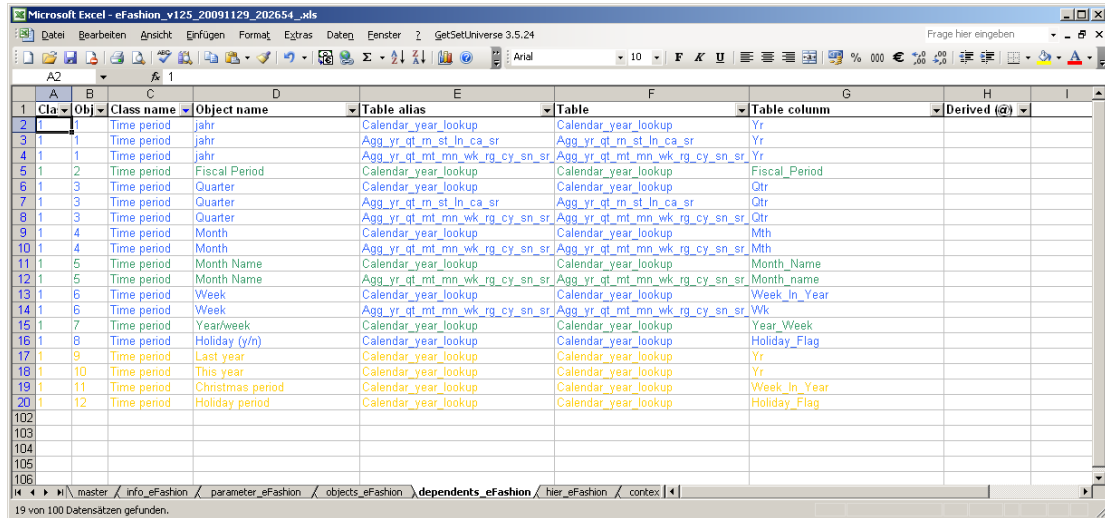


Pic 11 Advanced properties of an object

### 3.4 sheet dependents\_ <universe name>

The sheet dependents\_<universe name> contains an impact/lineage analysis. For each universe object the based table/ table column will be shown.

In case of a damaged data load all impacted objects can be identified with one click.



Pic 12 sheet dependents\_eFashion

Column	Meaning
Class ID	Id of class
Object ID	Id of object
Class name	Name of class
Object name	Name of object
Table alias	As the property select/ where of an object uses an alias table the name of the alias table will be shown. Otherwise the original name of the table will be shown.
Table	Name of the used database table in select/ where of the object
Table column	Name of the used database column in select/ where of the object
Derived	Flags objects containing @functions

#### Example 1

The condition ,Christmas period' (universe eFashion) is defined with SQL ,Calendar\_year\_lookup.Week\_In\_Year BETWEEN 46 AND 53'.

The object is documented in sheet dependents\_eFashion (row 19).

The object uses table Calendar\_year\_lookup and the column Week\_In\_Year.

For objects containing an @function the based objects and their table structure will be shown.

#### Example 2

The dimension object ,Month' (universe eFashion) is defined with SQL

,@aggregate\_aware( Agg\_yr\_qt\_mt\_mn\_wk\_rg\_cy\_sn\_sr\_qt\_ma.Mth, Calendar\_year\_lookup.Mth)'.

The object is documented in sheet dependents\_eFashion (rows 9 and 10).

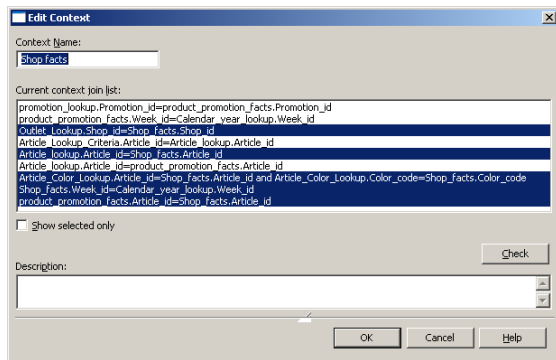
The object uses the table Calendar\_year\_lookup with the column Mth and the table Agg\_yr\_qt\_mt\_mn\_wk\_rg\_cy\_sn\_sr\_qt\_ma with the column Spalte Mth.

### 3.5 sheet contexts\_ <universe name>

The sheet contexts\_ <universe name> shows the defined contexts of the universe.

Context name	Table 1	Column 1	Table 2	Column 2	Expression
Shop facts	Outlet_Lookup	Shop_id	Shop_facts	Shop_id	outlet_lookup.shop_id=shop_facts.shop_id
Shop facts	Article_Lookup	Article_id	Shop_facts	Article_id	article_lookup.article_id=shop_facts.article_id
Shop facts	Article_Color_Lookup	Article_id	Shop_facts	Article_id	article_color_lookup.article_id=shop_facts.article_id and article_color_lookup.color_code=s
Shop facts	Article_Color_Lookup	Color_code	Shop_facts	Color_code	article_color_lookup.article_id=shop_facts.article_id and article_color_lookup.color_code=s
Shop facts	Shop_facts	Week_id	Calendar_year_lookup	Week_id	shop_facts.week_id=calendar_year_lookup.week_id
Shop facts	product_promotion_facts	Article_id	Shop_facts	Article_id	product_promotion_facts.article_id=shop_facts.article_id
Promotions	promotion_lookup	Promotion_id	product_promotion_facts	Promotion_id	promotion_lookup.promotion_id=product_promotion_facts.promotion_id
Promotions	product_promotion_facts	Week_id	Calendar_year_lookup	Week_id	product_promotion_facts.week_id=calendar_year_lookup.week_id
Promotions	Article_Lookup_Criteria	Article_id	Article_lookup	Article_id	article_lookup.criteria.article_id=article_lookup.article_id
Promotions	Article_lookup	Article_id	product_promotion_facts	Article_id	article_lookup.article_id=product_promotion_facts.article_id

Pic 13 sheet contexts\_eFashion



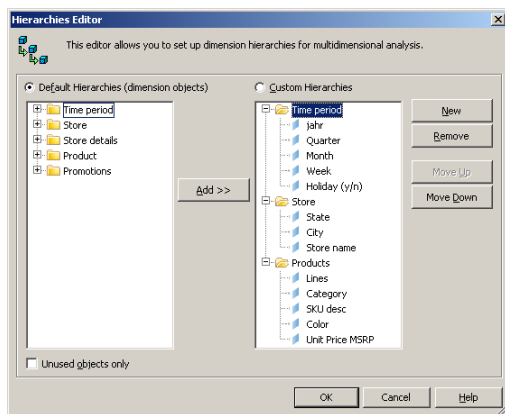
Pic 14 universe context menu

### 3.6 sheet hier\_<universe name>

The sheet hier\_<universe name> shows the defined hierarchies of the universe.

Hierarchy name	Object name	Class name	Hidden
Time period	Jahr	Time period	
Time period	Quarter	Time period	
Time period	Month	Time period	
Time period	Week	Time period	
Time period	Holiday (y/n)	Time period	
Store	State	Store	
Store	City	Store	
Store	Store name	Store	
Products	Lines	Product	
Products	Category	Product	
Products	SKU desc	Product	
Products	Color	Product	
Products	Unit Price MSRP	Product	

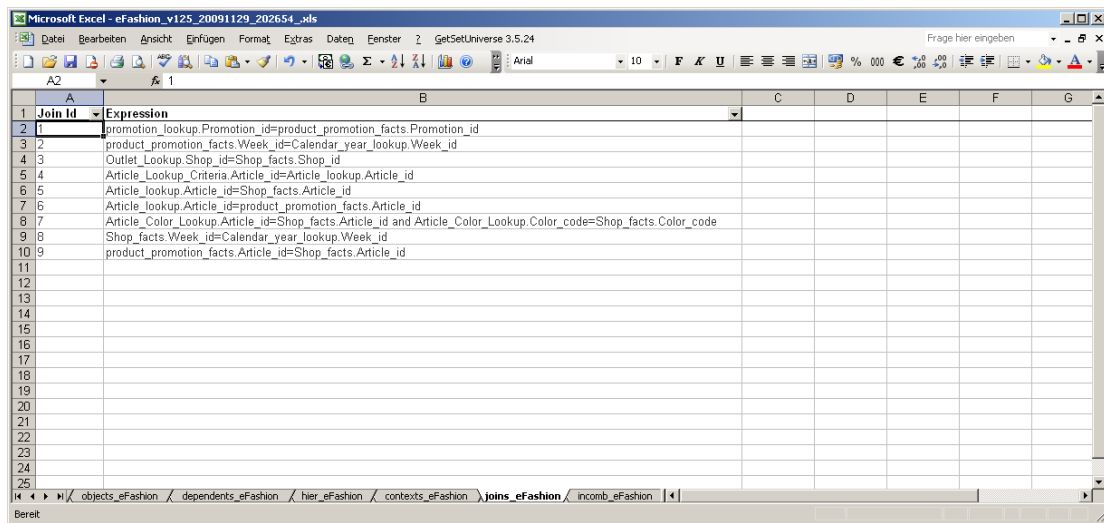
Pic 15 sheet hier\_eFashion



Pic 16 Universe hierarchies

### 3.7 sheet joins\_ <universe name>

The sheet joins\_ <universe name> shows the joins of the universe.



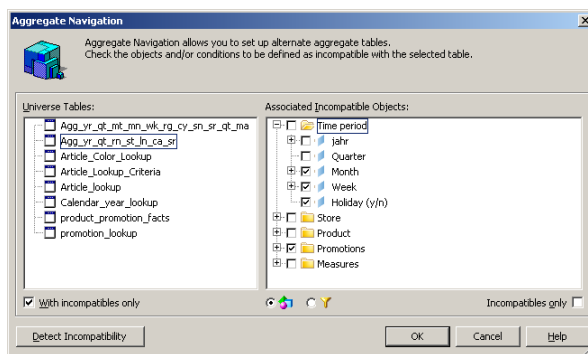
Pic 17 sheet joins\_eFashion

### 3.8 sheet `incomb_<universe name>`

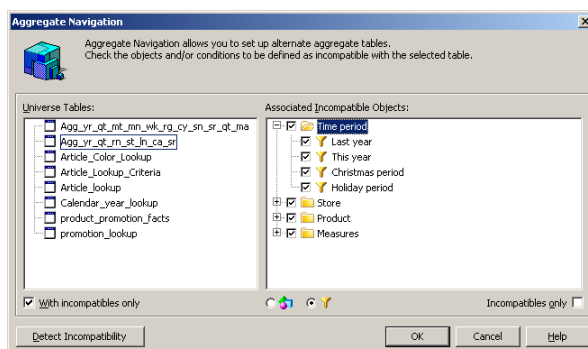
The sheet `incomb_<universe name>` documents the incompatibilities.

Table name	Class name	Object name	Table ID	Class ID	Object ID
Agg_yr_qt_m_st_in_ca_sr	Time period	Holiday (y/n)	9	1	8
Agg_yr_qt_m_st_in_ca_sr	Time period	Month	9	1	4
Agg_yr_qt_m_st_in_ca_sr	Time period	Year/week	9	1	7
Agg_yr_qt_m_st_in_ca_sr	Time period	Week	9	1	6
Agg_yr_qt_m_st_in_ca_sr	Time period	Month Name	9	1	5
Agg_yr_qt_m_st_in_ca_sr	Time period	Fiscal Period	9	1	2
Agg_yr_qt_m_st_in_ca_sr	Time period	This year	9	1	10
Agg_yr_qt_m_st_in_ca_sr	Time period	Last year	9	1	9
Agg_yr_qt_m_st_in_ca_sr	Time period	Christmas period	9	1	11
Agg_yr_qt_m_st_in_ca_sr	Time period	Holiday period	9	1	12

Pic 18 sheet `incomb_eFashion`

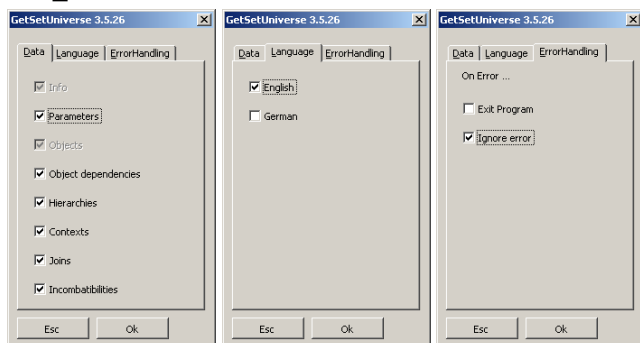


Pic 19 Universe Incompatibilities



Pic 20 Universe Incompatibilities

## 4 Options



Pic 21 Options

With Options the level of details can be managed.  
The selection of each option enables or disables the output of a related sheet.

Data	Meaning
Info	The generated excel sheet info_<Universe name> documents informations as name, version, creating user of the universe.
Parameters	The generated excel sheet parameter_<Universe name> documents all parameter of the universe
Objekts	The generated excel sheet objects_<Universe name> dokumentiert die Universenobjekte.
Object dependencies	The generated excel sheet dependents_<Universe name> documents the used tables/ alias tables and table columns. For Objects with @functions the based objects will be shown.
Hierarchies	The generated excel sheet hier_<Universe name> documents the defined hierarchies of the universe.
Kontexts	The generated excel sheet contexts_<Universe name> documents the defined contextx of the universe.
Joins	The generated excel sheet joins_<Universe name> documents the defined joins of the universe.
Incombatibilities	The generated excel sheet incom_<Universe name> documents the defined incombatibilities of the universe.

Language	Meaning
English	The menu items and the header of the outputs will be written in English.
German	The menu items and the header of the outputs will be written in German.

Error Handling	Meaning
Exit Program	In case of any error an error message will be generated. The message points to the program routine which raises the error. After that the program stops.
Ignore Error	All errors will be ignored.



## 5 Load Universe

With executing *Load Universe* you can load the universe objects into Excel (The data will be written into the document GetSetUniverse.xls). These data can be edited in Excel and reimported to the universe then.

The program *Load Universe* can run multiple times. With each run the extracted data in Excel will be overwritten. The loading of different universes is possible.

The program generates the following Excel sheets:

- info\_<universe name>
- objects\_<universe name>.

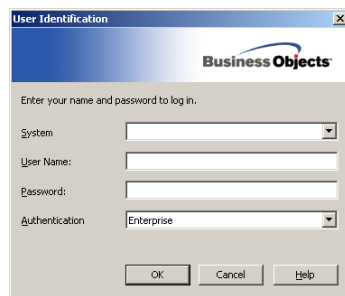
### 5.1 Process

#### 5.1.1 Run Load Universe

With selecting the item *Load Universe* the program will run.

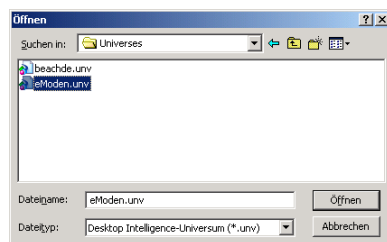
#### 5.1.2 Log on as Designer

The log on to the Business Objects Designer is required.



#### 5.1.3 Select universe

Select the universe you will load.



#### 5.1.4 Result

The extracted universe data will be stored into the sheets info\_<universe name> and objects\_<universe name>. All names of the sheets will be used for processing the data. Changing the names of the sheets will result errors.

## 6 Write Universe

With a successful import of the universe data (by using the program *Load Universe*) the following properties are editable:

- Object name
- Description
- SQL Select
- SQL Where.

These properties are stored in the sheet objects\_<universe name>.

The modified properties can be rewritten into the universe by running the program *Write Universe*.

Only the local stored version of the universe will be processed.

The universe can be exported into the repository by the designer. An automatic export of the universe is not supported by GetSetUniverse!

### 6.1 Process

#### 6.1.1 Preconditions

First precondition for modifying the universe data is their import into Excel (see Load Universe)

For correct processing of Write Universe it is necessary to leave the sort sequence of the data in sheet objects\_<universe name>. Otherwise the mapping of changes to the current definitions is not possible. In case of changes in the sort sequence the universe should be loaded again into Excel (see Load Universe).

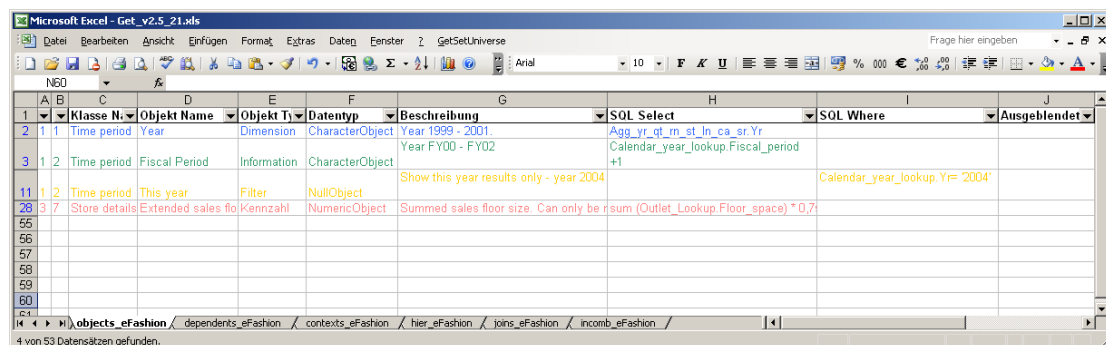
#### 6.1.2 Changing the object properties

The following properties (of an object) are editable:

Object type	Object name	Description	SQL Select	SQL Where
Dimension	X	X	X	X
Detail	X	X	X	X
Measure	X	X	X	X
Condition	X	X		X

Only the 4 columns Object name, Description, SQL Select and SQL Where are editable.

As shown in *PIC 22 Write Universe, Changing of object definitions* the properties has been changed exemplary for each type of objects.



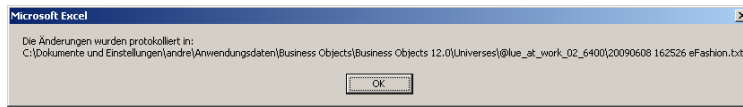
PIC 22 Write Universe, Changing of object definitions

#### 6.1.3 Start Write Universe

By running <GetSetUniverse ><Write Universe> the changed data will be written back into the universe.

### 6.1.4 Log of changes

All changes will be logged. With completion of *Write Universe* the message refers to the protocol.



Pic 23 Write Universe, Log reference

The file name of the log contains the time stamp(processing time <day><time>) and the universe name.

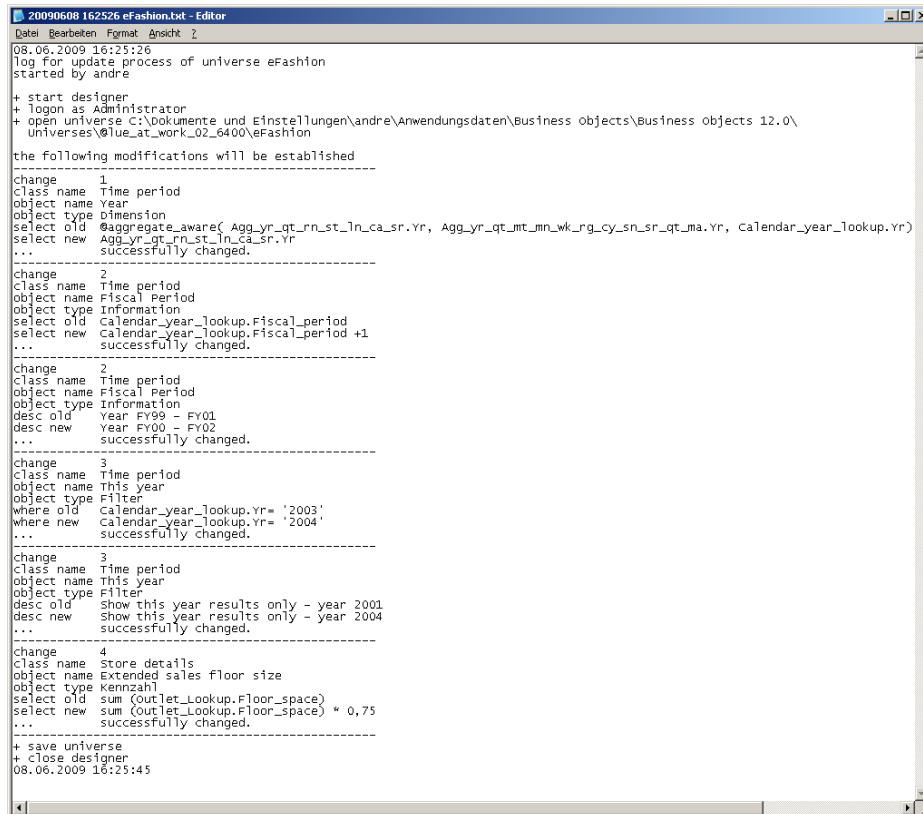


Abbildung 24 Schreibe Universum, Protokoll

The log documents

- time stamp of start
- BO user name
- Log on data
- Folder and name of used local universe
- Status of modifications
- Object
  - ▣ property before
  - ▣ property after

### Error messages

For a condition no SQL Select is definable. In case of modifying the property SQL Select for a condition an adequate error message will appear.

### 6.1.5 Check the results

The resulting universe must be checked by opening in within Business Objects Universe Designer and run the internal check routines. If everything is fine, the universe can be exported then.

## 7 Versioning Universe

*Versioning Universe* loads the universe data within an additional Excel file.

The name of the version file contains

- the universe name,
- the versions of the universe and
- the time stamp of the versioning process.

The file *eFashion\_v125\_20090608\_164748.xls* documents the 125. version of the universe *eFashion*. The version has been created on 08.06.2009 16:47:48.

### 7.1 Process

#### 7.1.1 Auswahl Optionen

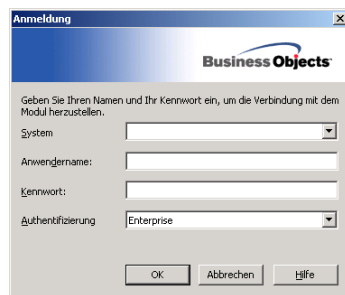
The level of detail (see Options) defines which universe data will be written to the version file.

#### 7.1.2 Start Versioning Universe

Select the item *Versioning Universum* to run the program.

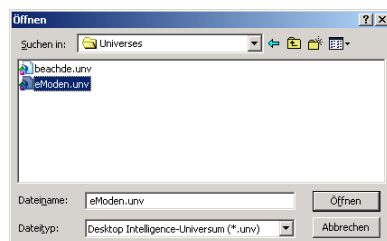
#### 7.1.3 Log on as Designer

Log on as Designer.



#### 7.1.4 Select the Universe

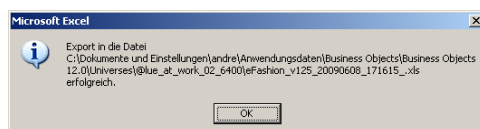
Select the Universe.



The program loads the universe data and stores them in the assigned excel sheets *info\_<universe name>*, etc..

## 7.2 Result

With completion of the program the message points out the folder in which the version is stored.



By using diff tools (as Winmerge) data of different versions can be compared.

## 8 Error handling

In case of an error the error message (containing the name of the routine, which raises the error) will be written.

### 8.1 Error during reading the universe data

As the error message points to one of the following routines, the error occurs during the reading of the universe.

Routinen
GetUniverseClasses
GetUniverseSubClasses
GetUniverseParameter
GetUniverseTables
GetUniverseTableColumns
GetUniverseContexts
GetUniverseJoins
GetUniverseHierarchies

This can happen if the program will read a universe in a higher version than expected (see.1 Summary). As the SDK can vary between the versions, properties of universe data (of a higher BO version) can not be identified by *GetSetUniverse*.

As work around the reading of those properties can be ignored by changing the option `ErrorHandling` from `<Exit Program>` to `<Ignore Error>` (see. 4 Options).

### 8.2 Additional Errors

All additional errors are unknown.