

Managing Multiple Languages

In Business Objects Universes

Version 1.0

14th February 2003

Author: Peter Nolan

Peter@peternolan.net

Table of Contents

1. CHANGE CONTROL LOG.....	3
2. AUDIENCE	4
3. OVERVIEW.....	4
4. ASSUMPTIONS IN THIS DOCUMENT	4
5. INTRODUCTION	5
6. RESEARCH.....	5
6.1. Translation Tables + Updated Universe.....	5
7. IMPLEMENTED EXAMPLE	6
7.1. PROCESS OF TESTING MULIT LANGUAGE	6
7.1.1. SQL To Update Classes/Objects.....	10
8. ISSUES.....	18

1. **CHANGE CONTROL LOG**

#	Date	Name	Description
1.0	14/2/2003	P. Nolan	Initial version for publication on my web page.

2. AUDIENCE

The intended audiences for this document are:

- Business Objects Designers who want to manage multiple language reports using Business Objects.

3. OVERVIEW

The purpose of this document is to document the results of implementing multi-language universes in Business Objects. This document is limited in scope to discuss ways in which to implement multi-language universes in Business Objects.

If you would like to ask questions please direct them to peter@peternolan.net.

Please note. If you would like the MS Access database and reports used to create this document please contact me on peter@peternolan.net.

4. ASSUMPTIONS IN THIS DOCUMENT

For the remainder of the document I will make the following assumptions: I assume:

- You are very familiar with the Business Objects universe data model and the data that is stored in that data model.
- You are familiar with how to write Business Objects reports.
- You are familiar with the SQL language and understand how to use SQL to update tables.

5. INTRODUCTION

I recently did a Sybase IWS/Business Objects project where one of the Business Requirements for the implementation was to enable multiple languages so that staff in Norway, Sweden, Finland and Denmark were all able to access the IWS in their own language as well as English.

We planned to implement the IWS first in English and then roll out the different languages as they were perceived to be needed.

We did quite a bit of research into how this might be possible. This document contains some details of the best result we were able to produce.

6. RESEARCH

Research on this topic proved relatively difficult as we were unable to find any individual who has performed such an implementation with Business Objects. I believe the solution documented here to be the best solution we could find. If anyone else reading this document has a better idea I would love to hear it!!

6.1. Translation Tables + Updated Universe

This section describes the best option which is to do the following:

1. Store a set of translation tables in the BO universe database.
2. In Business Objects develop a 'primary' universe in English.
3. In business Objects develop a 'secondary' universe in Norwegian/Swedish.
4. Further languages would require the development of further universes.
5. Write SQL that will use the English IWS Universe as the 'primary' universe and using the English universe populate a number of 'secondary' universes translating the English to the other language using the translation table.

7. IMPLEMENTED EXAMPLE

This section documents an example of managing multi-language universes within Business Objects.

Multi-Language support as I have been able to prove consists of being able to present N versions of the universe and associated reports.

7.1. PROCESS OF TESTING MULTILANGUAGE

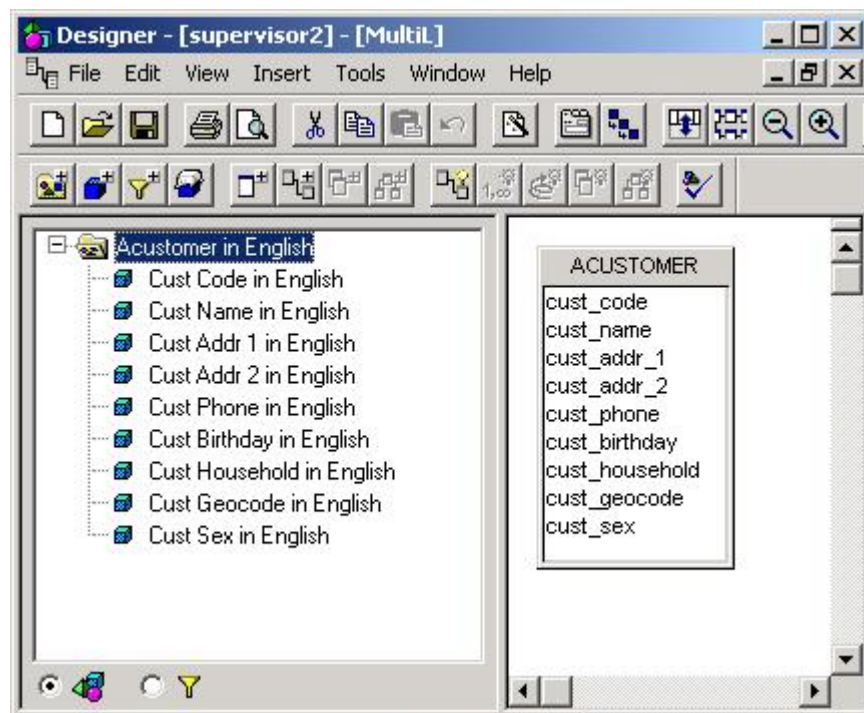
This section documents an actual test case of doing what I thought was possible.

Step 1

Create a new repository in BO.

Step 2

Create a universe (MultiL) with one table (Acustomer)



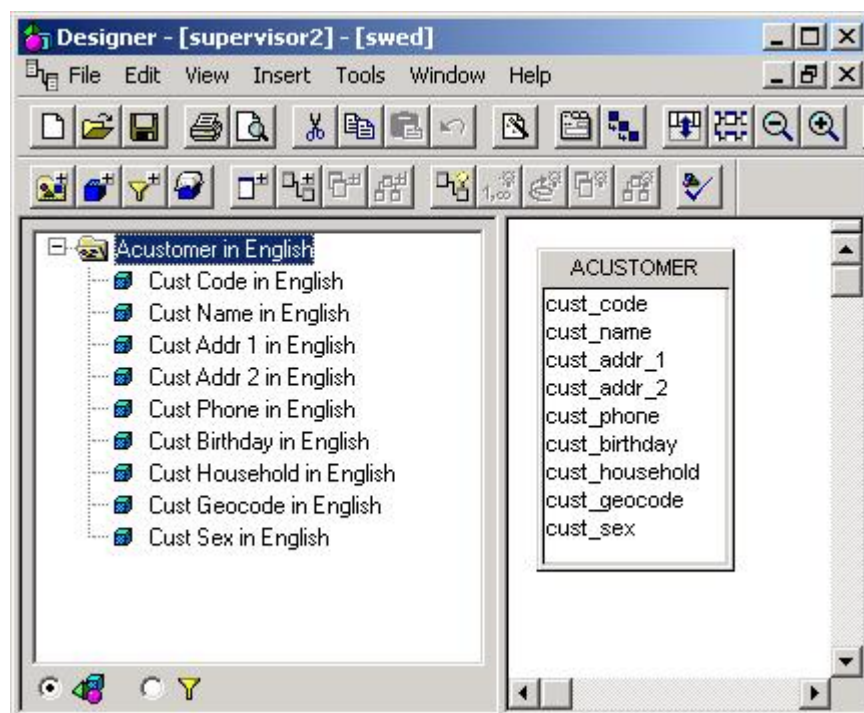
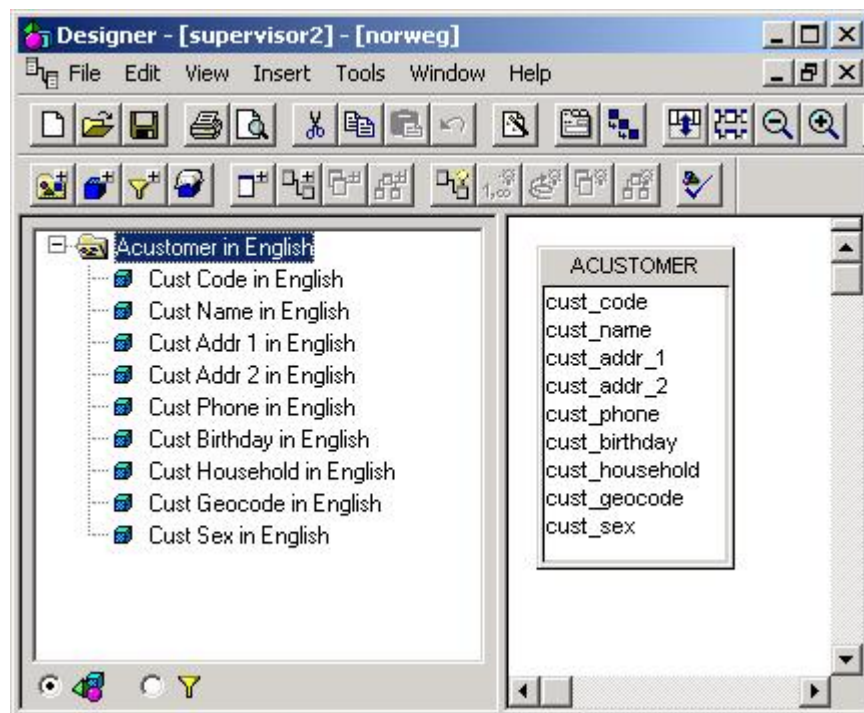
Step 3

Note that you should change the name of the object and class. In this case I have put 'in English' on each class and object to distinguish that this is the 'English' version.

Note also that I added help text to the class and to each object. The language translator updates rows. If there are no help text rows for English it will not insert a row for any other language.

Step 4

Then save the English (MultiL) universe as norwegian/swed as appropriate, change the parameter name as well and create two 'slave' universes. When you export them to the repository they will look exactly the same as the MultiL universe.



Step 5

Set up two tables to test the translation of the text fields that are displayed on the reports and the help text.

W_TRANSLATE_TABLES

SRC_UNIVERSE_ID	INTEGER
SRC_OBJECT_ID	INTEGER
SRC_TABLE_NAME	VARCHAR 30
SRC_CLASS_NAME	VARCHAR 30
NOR_UNIVERSE_ID	INTEGER
NOR_CLASS_NAME	VARCHAR 30
NOR_HELP_TEXT	VARCHAR 255
SWE_UNIVERSE_ID	INTEGER
SWE_CLASS_NAME	VARCHAR 30
SWE_HELP_TEXT	VARCHAR 255

W_TRANSLATE_COLS

SRC_UNIVERSE_ID	INTEGER
SRC_OBJECT_ID	INTEGER
SRC_CLASS_NAME	VARCHAR 30
SRC_OBJECT_NAME	VARCHAR 30
NOR_UNIVERSE_ID	INTEGER
NOR_OBJECT_NAME	VARCHAR 30
NOR_OBJECT_HELP_TEXT	VARCHAR 255
SWE_UNIVERSE_ID	INTEGER
SWE_OBJECT_NAME	VARCHAR 30
SWE_OBJECT_HELP_TEXT	VARCHAR 255

These tables have the object_ID of the table or column at the front of them to identify the specific object to be translated in target universe.

The picture below shows an example of w_translate_tables.

src_u	src_table_name	src_class_name	nor_u	nor_class_name	nor_help_text	swe_u	swe_class_name
1	ACUSTOMER	Acustomer in English	12	Acustomer in Norwegian	Help text in Norwegian	13	Acustomer in Swedish

The picture below shows an example of w_translate_columns.

src_object_name	nor_u	nor_object_name	nor_help_text	swe_u	swe_object_name
Cust Code in English	12	Cust Code in Norwegian	Help Cust Code in Norwegian	13	Cust Code in Swedish
Cust Name in English	12	Cust Name in Norwegian	Help Cust Name in Norwegian	13	Cust Name in Swedish
Cust Addr 1 in English	12	Cust Addr 1 in Norwegian	Help Cust Addr 1 in Norwegian	13	Cust Addr 1 in Swedish
Cust Addr 2 in English	12	Cust Addr 2 in Norwegian	Help Cust Addr 2 in Norwegian	13	Cust Addr 2 in Swedish
Cust Phone in English	12	Cust Phone in Norwegian	Help Cust Phone in Norwegian	13	Cust Phone in Swedish
Cust Birthday in English	12	Cust Birthday in Norwegian	Help Cust Birthday in Norwegian	13	Cust Birthday in Swedish
Cust Household in English	12	Cust Household in Norwegian	Help Cust Household in Norwegian	13	Cust Household in Swedish
Cust Geocode in English	12	Cust Geocode in Norwegian	Help Cust Geocode in Norwegian	13	Cust Geocode in Swedish
Cust Sex in English	12	Cust Sex in Norwegian	Help Cust Sex in Norwegian	13	Cust Sex in Swedish

Step 6

In this step you must write the SQL scripts that perform the translation of the secondary repository. This will depend on the database in which you are storing your business objects universe and the translate tables. I have put the test versions in an MS Access database. So the SQL reproduced below is the MS Access version.

7.1.1. SQL To Update Classes/Objects

```
UPDATE w_translate_tables INNER JOIN UNV_CLASS ON
([w_translate_tables].[src_object_id]=[UNV_CLASS].[CLASS_ID]) AND
([w_translate_tables].[nor_universe_id]=[UNV_CLASS].[UNIVERSE_ID]) SET UNV_CLASS.CLS_NAME =
[w_translate_tables].[nor_class_name];

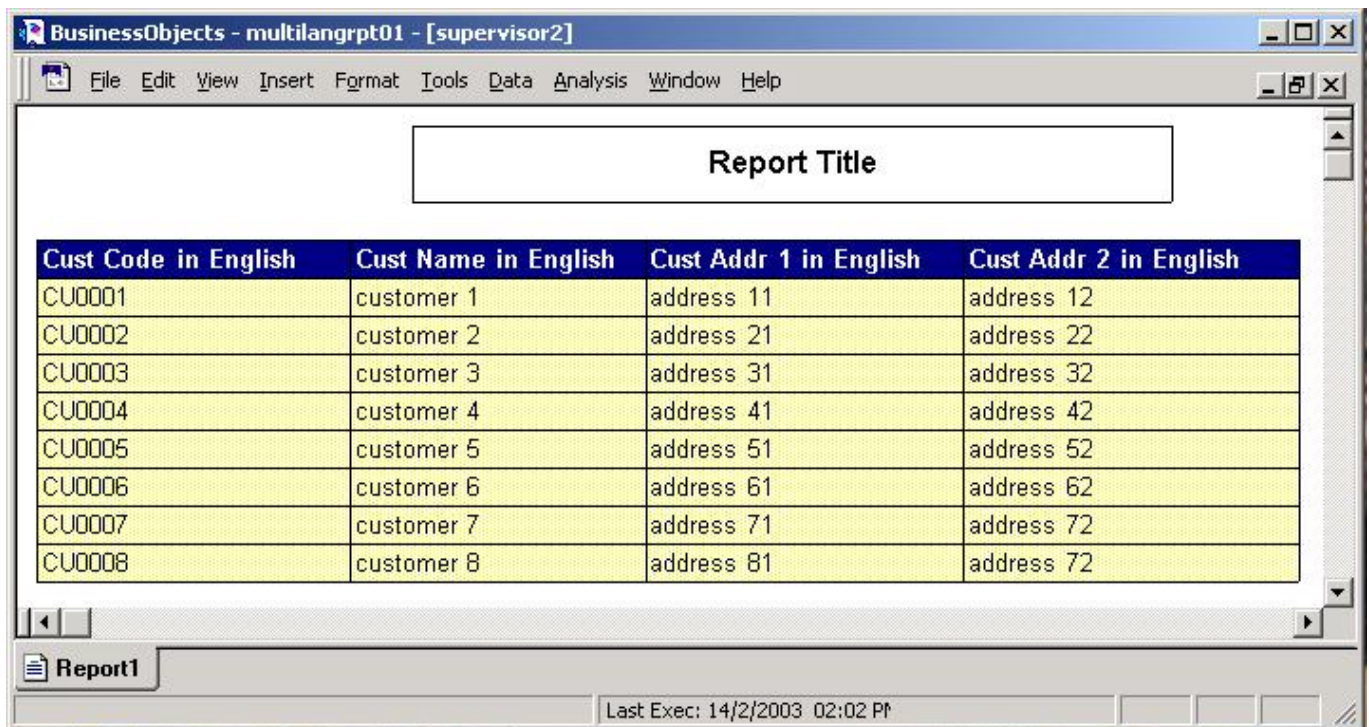
UPDATE w_translate_tables INNER JOIN UNV_CLASS_DATA ON (w_translate_tables.nor_universe_id =
UNV_CLASS_DATA.UNIVERSE_ID) AND (w_translate_tables.src_object_id = UNV_CLASS_DATA.CLASS_ID)
SET UNV_CLASS_DATA.CLS_DATAVALUE = [w_translate_tables].[nor_help_text]
WHERE (((UNV_CLASS_DATA.CLS_DATATYPE)="H"));

UPDATE w_translate_cols INNER JOIN UNV_OBJECT ON (w_translate_cols.nor_universe_id =
UNV_OBJECT.UNIVERSE_ID) AND (w_translate_cols.src_object_id = UNV_OBJECT.OBJECT_ID) SET
UNV_OBJECT.OBJ_NAME = [w_translate_cols].[nor_object_name];

UPDATE w_translate_cols INNER JOIN UNV_OBJECT_DATA ON (w_translate_cols.src_object_id =
UNV_OBJECT_DATA.OBJECT_ID) AND (w_translate_cols.nor_universe_id =
UNV_OBJECT_DATA.UNIVERSE_ID) SET UNV_OBJECT_DATA.OBJ_DATAVALUE =
[w_translate_cols].[nor_help_text]
WHERE (((UNV_OBJECT_DATA.OBJ_DATATYPE)="H"));
```

Step 7

Create a report called multilangrpt01 which is just the first few fields from the ACUSTOMER table using the MultiL universe.



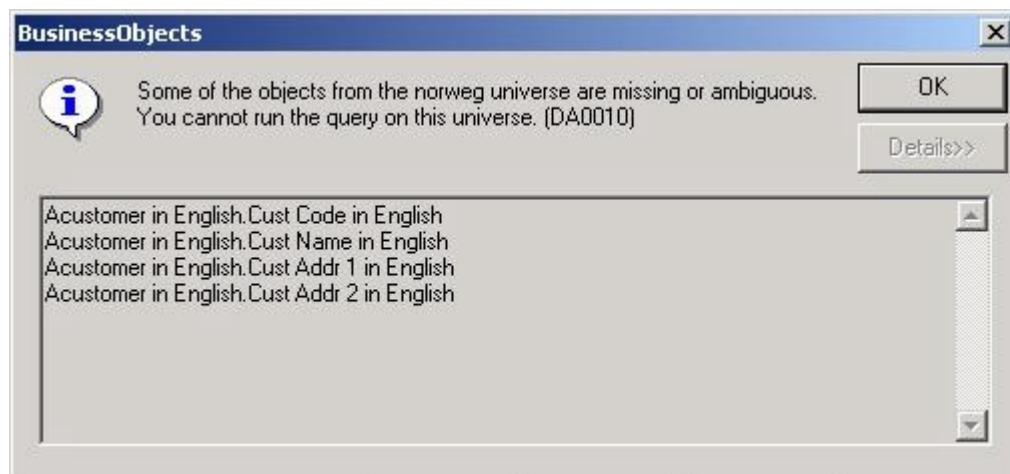
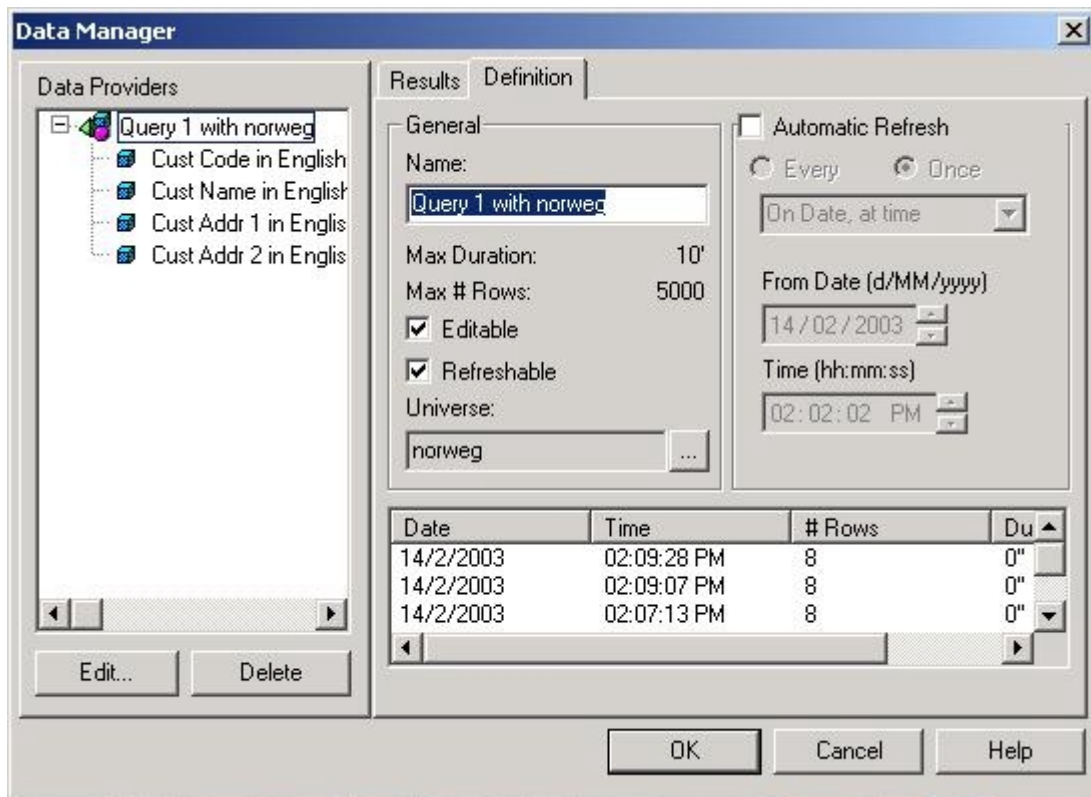
The screenshot shows a BusinessObjects report viewer window titled "BusinessObjects - multilangrpt01 - [supervisor2]". The menu bar includes File, Edit, View, Insert, Format, Tools, Data, Analysis, Window, and Help. The report content features a "Report Title" box at the top, followed by a table with four columns: "Cust Code in English", "Cust Name in English", "Cust Addr 1 in English", and "Cust Addr 2 in English". The table contains eight rows of data. At the bottom, there is a "Report1" tab and a status bar indicating "Last Exec: 14/2/2003 02:02 PM".

Cust Code in English	Cust Name in English	Cust Addr 1 in English	Cust Addr 2 in English
CU0001	customer 1	address 11	address 12
CU0002	customer 2	address 21	address 22
CU0003	customer 3	address 31	address 32
CU0004	customer 4	address 41	address 42
CU0005	customer 5	address 51	address 52
CU0006	customer 6	address 61	address 62
CU0007	customer 7	address 71	address 72
CU0008	customer 8	address 81	address 72

Step 8

Don't do this!!

Note that if you try to convert the language of the universes prior to converting the language of the reports you will see the following happen.



What this means is that Business Objects checks the objects on a report at the name level when changing from one universe to another. Fair enough.

To fool Business Objects you have to change the universe used to run the query while it still has the same column names. This can probably be done by setting up a test universe that is a direct copy of the English universe on a single test machine and converting all reports to the other language universe (say Norwegian or Swedish) before publishing the universe to the real Norwegian universe.

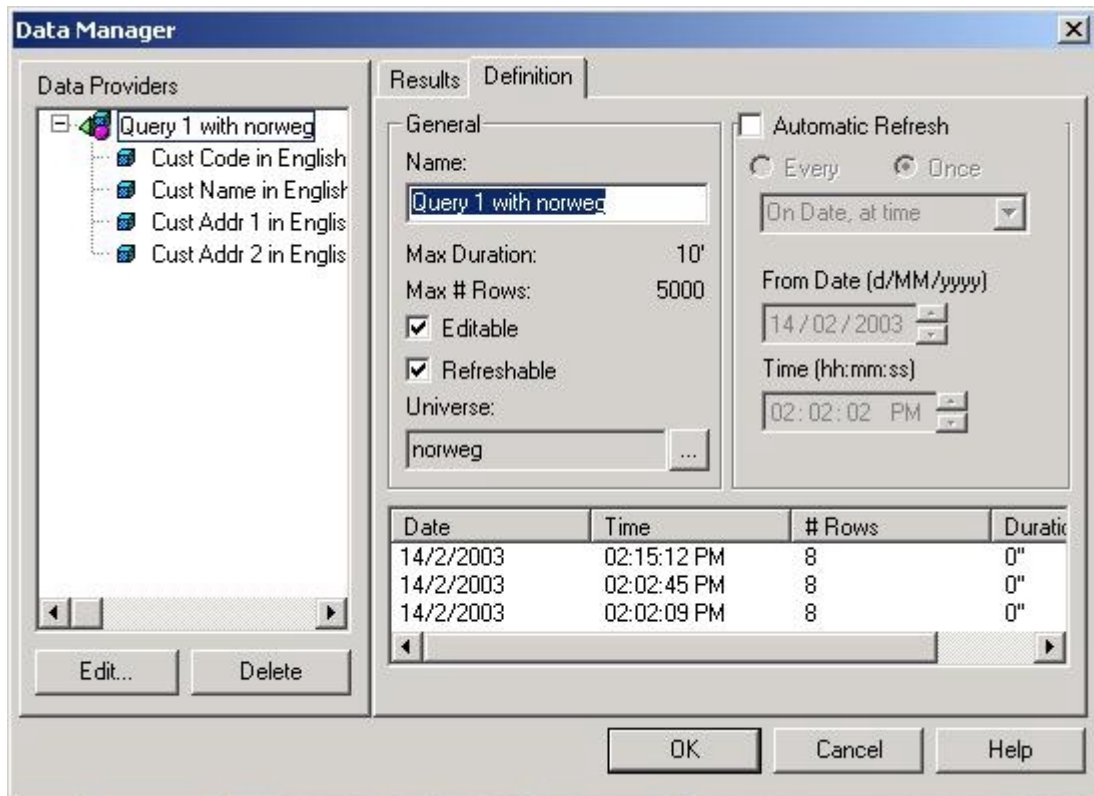
Step 9

So, to get around the problem mentioned above, you need to create your other language reports, and connect them to their other language universes before converting the universes. I created a Multilanguage report and changed the universe to use before changing the language of the universe.

Step 10

Once this has been done, I:

- Changed the language of the Norwegian and Swedish universe
- Re-import the universes to the desktop
- Opened the report
- Went into view data -> data manager
- Clicked on 'refresh'



And the object names will be 'magically' changed to the new language column names. Business objects refreshes the text of the objects on the report from the repository without checking the names themselves because the name of the universe has not changed. It seems to update them based on object id.

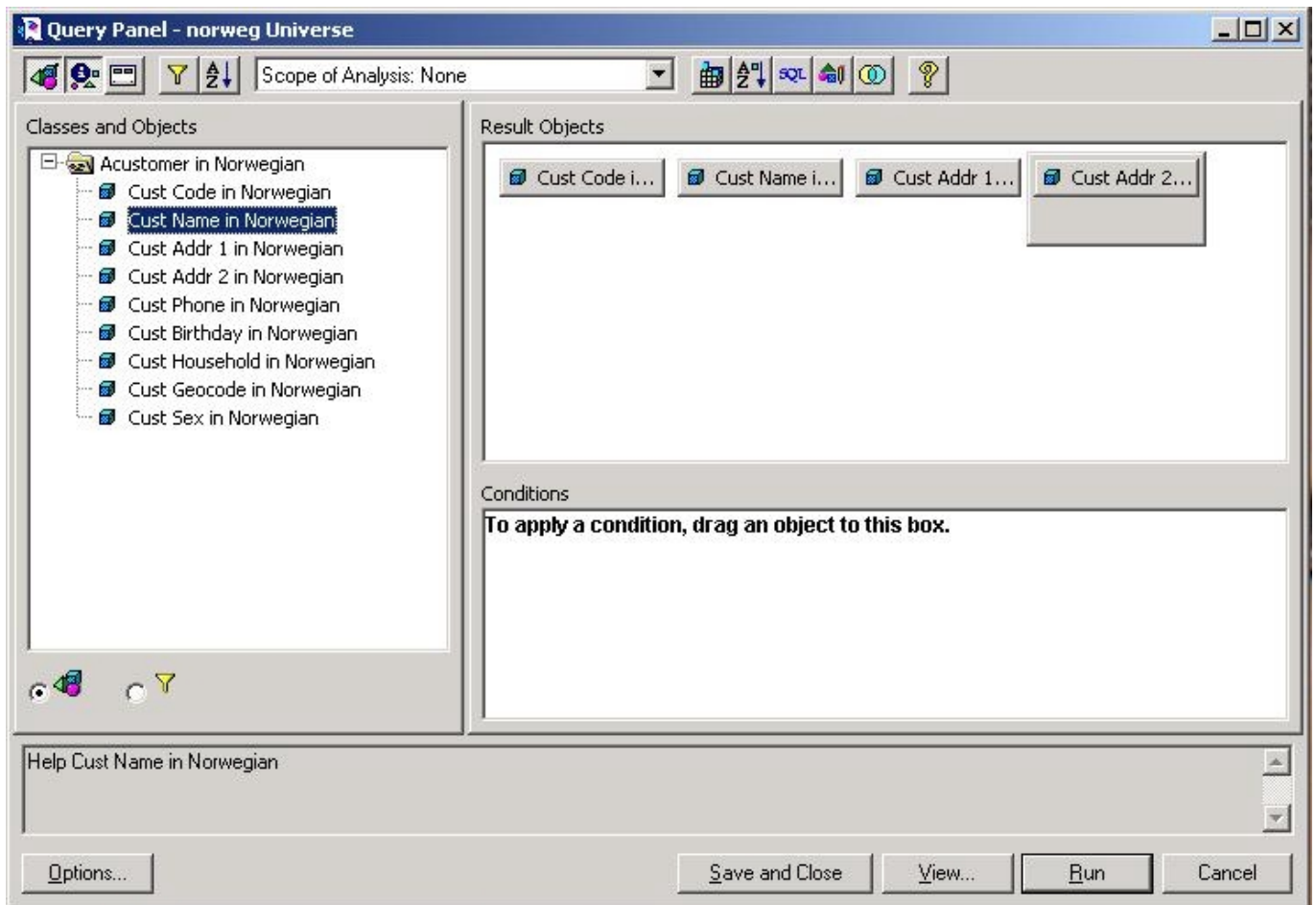
The report then changes to look as follows:

Cust Code in Norwegian	Cust Name in Norwegian	Cust Addr 1 in Norwegian	Cust Addr 2 in Norwegian
CU0001	customer 1	address 11	address 12
CU0002	customer 2	address 21	address 22
CU0003	customer 3	address 31	address 32
CU0004	customer 4	address 41	address 42
CU0005	customer 5	address 51	address 52
CU0006	customer 6	address 61	address 62
CU0007	customer 7	address 71	address 72
CU0008	customer 8	address 81	address 72

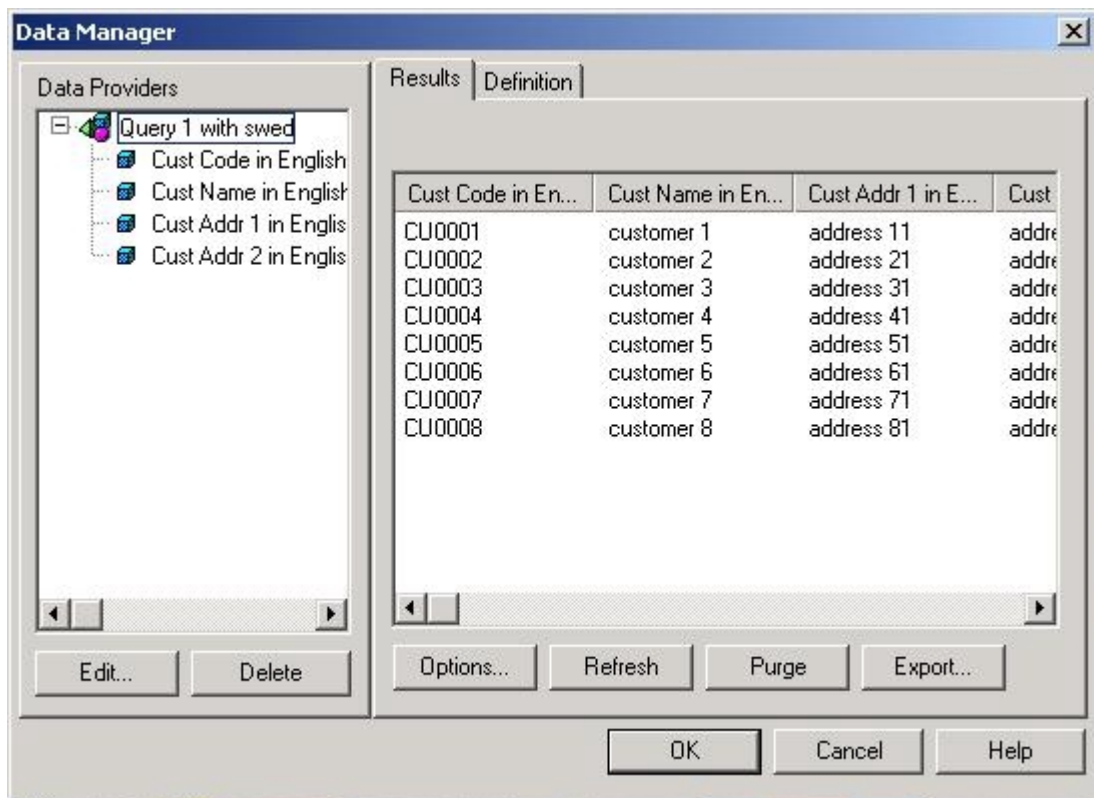
Report1

Last Exec: 14/2/2003 02:18 PM

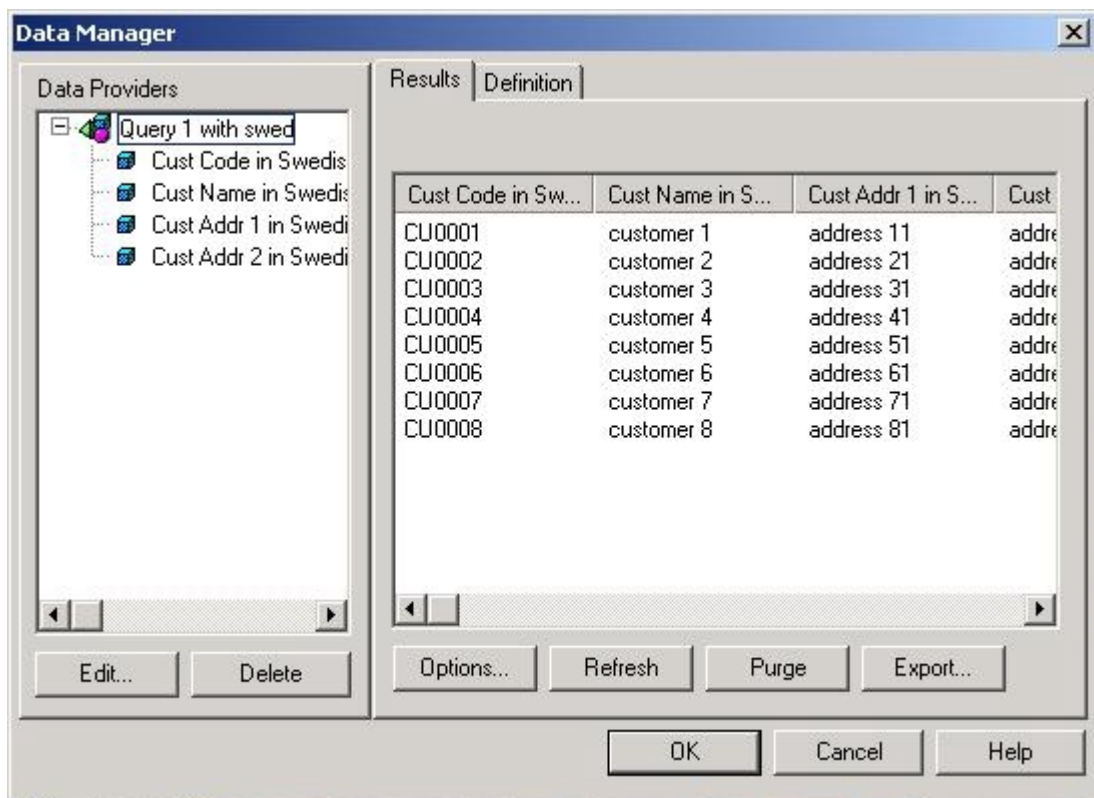
Notice also that the help text has changed to be the Norwegian help text.



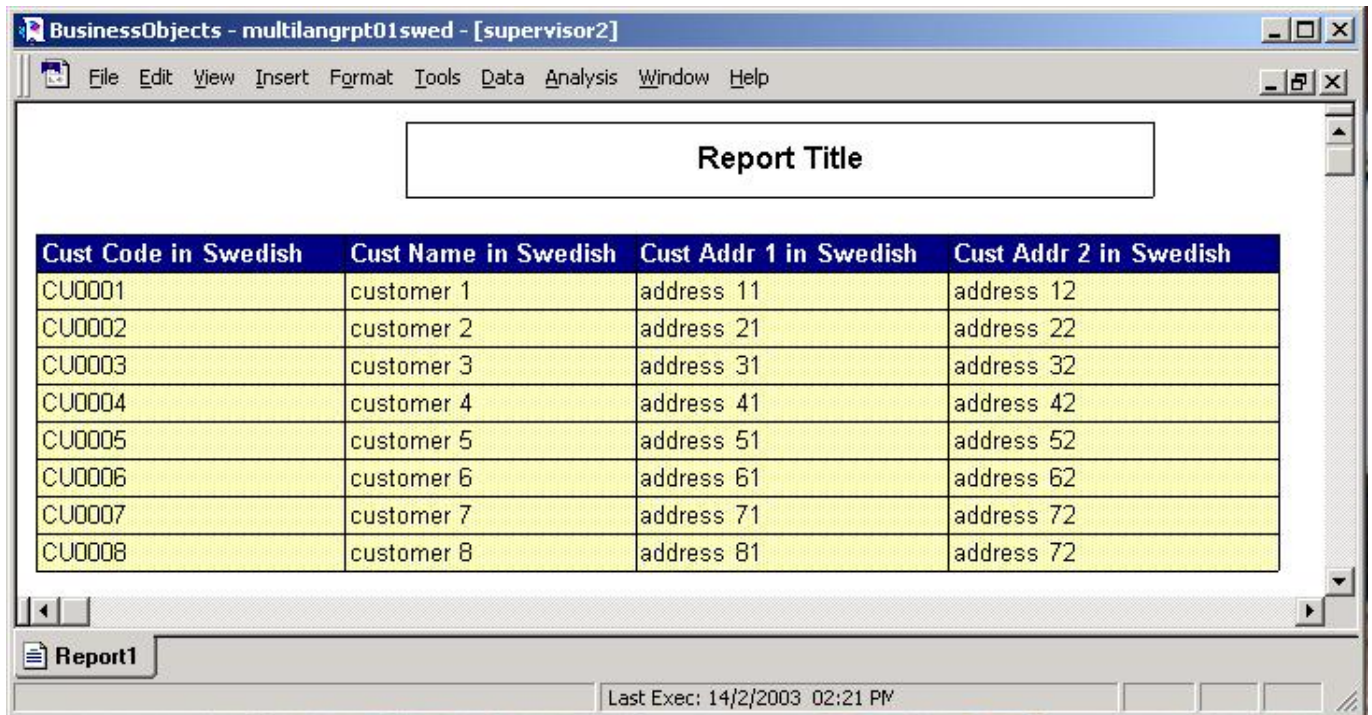
To see this in a little more detail let's look at doing the same for the Swedish report. First, we open the view data-> data manager window. And we see the data manager, though using the swed universe, still has the English names.



We click on the 'Refresh' button and the names change to the 'Swedish' names. Pretty cool when you think about it.



The report then looks like the object names have changed to Swedish.



Cust Code in Swedish	Cust Name in Swedish	Cust Addr 1 in Swedish	Cust Addr 2 in Swedish
CU0001	customer 1	address 11	address 12
CU0002	customer 2	address 21	address 22
CU0003	customer 3	address 31	address 32
CU0004	customer 4	address 41	address 42
CU0005	customer 5	address 51	address 52
CU0006	customer 6	address 61	address 62
CU0007	customer 7	address 71	address 72
CU0008	customer 8	address 81	address 72

8. ISSUES

This section lists the aspects of this multi-language management that I expect will be perceived as issues.

I1. Mapping table.

It is necessary to store text in a number of tables to map the 'primary' language to the 'secondary' language repositories. The tables need to store all class names, all object names for all objects in all universes. This requires some form of application to regularly update this table. No report can be converted to multiple languages unless this table is maintained. This is a maintenance role for one of the BO support staff.

I2. All objects in ALL Languages

This solution makes it necessary to make sure all objects are available in all languages. This means that if someone in Sweden, say, wanted a new object to be created specifically for Sweden, then that object must be created for all other languages regardless of the fact that it may not be required. This is because all mappings between languages is done on the basis of the object number and the object numbers must be consistent across all universes. All universes must be created from a 'master' universe to make sure the object numbers are the same across all universes. Business Objects does not seem to like 'deleted objects' so it is recommended no objects are deleted once they are created. They should be hidden rather than deleted.

I3. Staging Universes

It is not very feasible to just have a source and a number of target universes. Business Objects checks the fields in a report at class name/object name level when the universe used to produce that report is changed. Hence, when the universe to produce a report is changed the target universe must have the same column names. Since you do not want to be converting object names back and forth between languages you will require target universes which are in a specific language. The target universe will always be in that language. You would then propagate changes into the target universe and convert the language as quickly as possible so the 'primary' language is rarely seen by users in the target universe.

This means when changes are made to a universe following process is recommended:

1. The changes are pushed to the N staging universes.
2. Any new reports are pushed to the N staging universes,
3. The reports are run to update the object names
4. The 'primary' universe is exported to each of the N language universes.
5. The N language universes are converted to each of their languages.
6. The reports are then pushed from the staging universes down to the N language universes.

This increases the complexity of publishing the reports in the N languages.

These are the major issues. If any more issues arise they will be documented in a new version of the newsletter.

I4. No Translation for Report Text.

Text on the report, such as report title, can not be translated using this translator.

If Report Titles are to be translated they would need to be selected from a database which is country number sensitive to select the correct report heading text. I see this as a more minor issue mostly because I don't believe there is much that can be done about it.