

SPARX SYSTEMS ENTERPRISE
ARCHITECT GOES CLOUD
USING MICROSOFT AZURE AND
MICROSOFT SQL AZURE

VERSION 2.1

© 2012
LieberLieber Software
GmbH
www.lieberlieber.com

INHALT

Sparx Systems Enterprise Architect goes Cloud USING Microsoft Azure And Microsoft SQL Azure	2
Executive Summary	2
Preperations	2
Step by step Assistance	2
Step 1: Convert EAP File into Access 2010 Format.....	2
Step 2: Create a new database in SQL Azure	3
Step 3: Microsoft SQL Server Migration Assitant for Access (SSMA)	3
Step 4: Open SQL azure based Repository of Enterprise Architect.....	7
Results	8
Some Hints	10
Hint 1: database Connection	10
Hint 2: Being Inactive with EA	10
Hint 3: Fast Internet Connection	10
More Tips.....	10
Tip 1: Performance optimization.....	10
Tip 2: Working in TEAMS with Enterprise Architect	11
Tip 3: Use of replication and data synchronisation.....	11
End of Document.....	11

SPARX SYSTEMS ENTERPRISE ARCHITECT GOES CLOUD USING MICROSOFT AZURE AND MICROSOFT SQL AZURE

Author: Peter Lieber, May-June 2012

EXECUTIVE SUMMARY

This whitepaper describes how to migrate and run an Enterprise Architect Repository in the Microsoft Azure Cloud – esp. Microsoft SQL Azure.

NEW (Version 2.0): And additionally for Version 2.0 there were added important notes what will work and what will just look like it will work.

PREPERATIONS

- Enterprise Architect \geq 9.3 (Corporate Edition or higher)
- An Enterprise Architect Repository, prepared as EAP File (EAP = Microsoft Access Format) → in this document: *EAExample.eap*
- Microsoft Access \geq 2010
- Microsoft SQL Server Migration Assitant for Access (SSMA) \geq 5.2
- Microsoft Azure Abonnement \geq 1.6
- Microsoft SQL Azure Instance \geq 1.6
- NEW (Version 2.0): SQL Server Native Client \geq 10.0
- NEW (Version 2.0): as an attractive, alternative solution: CNS Media Gateway (www.connecting-software.com)

STEP BY STEP ASSISTANCE

STEP 1: CONVERT EAP FILE INTO ACCESS 2010 FORMAT

As a first step open the EAExample.eap file using Microsoft Access 2010 (right mouse button in Windows Explorer on file and „Open with...“ – there choose Microsoft Access 2010)

Access will inform you that the EAP is created by an older version of Access – choose yes to convert the file into a newer format and give another name.

During Conversion there will be 1 conversion issue and 1 hint that security on users level is not supported any more – this can be ignored and the created conversion issue table can be deleted or ignored.

STEP 2: CREATE A NEW DATABASE IN SQL AZURE

For this whitepaper we mention that a Microsoft Azure Abonnement is already existing and also the access to the administration portal: <https://windows.azure.com>

In our sample the name of the SQL Azure database server is: *ecmfpl512j.database.windows.net*. The database is called *EAExample* and configured as *1GB* size, type: *WEB*.

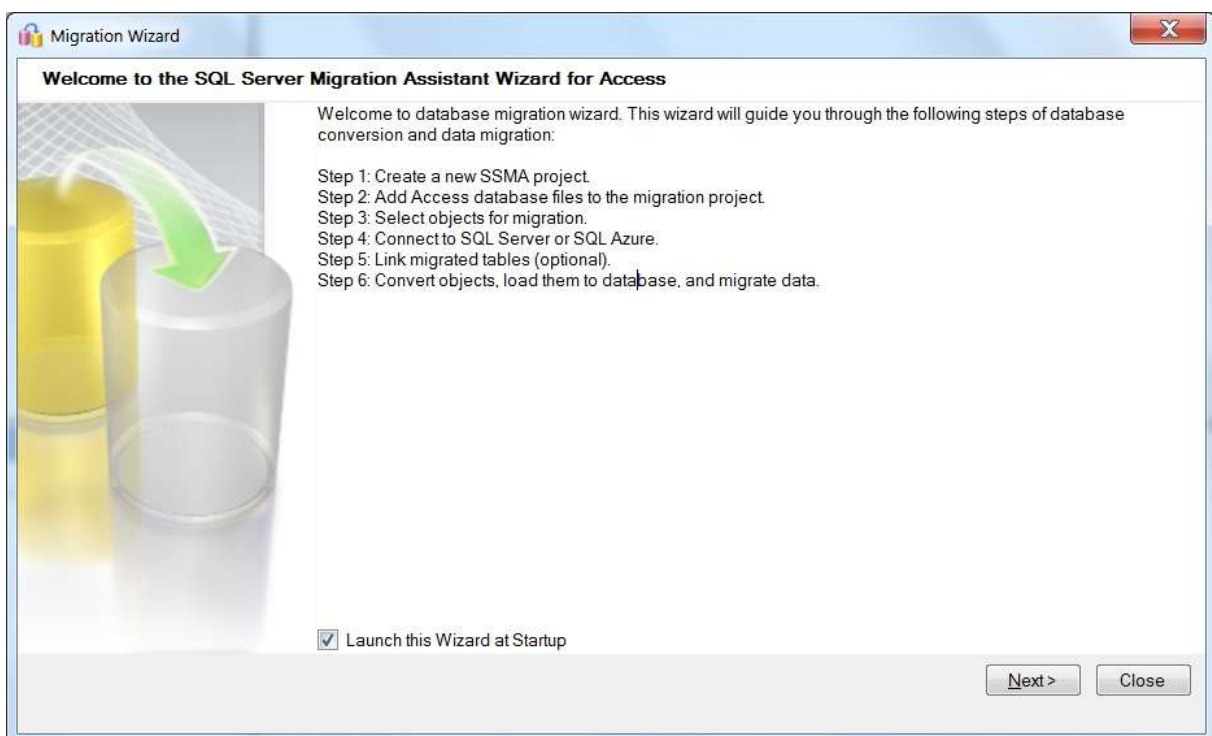
The administrative user is: *EAExampleAdmin* and the password: *Eee2012\$App\$E*.

(Hint: This database is not public available and only used for demonstration purposes)

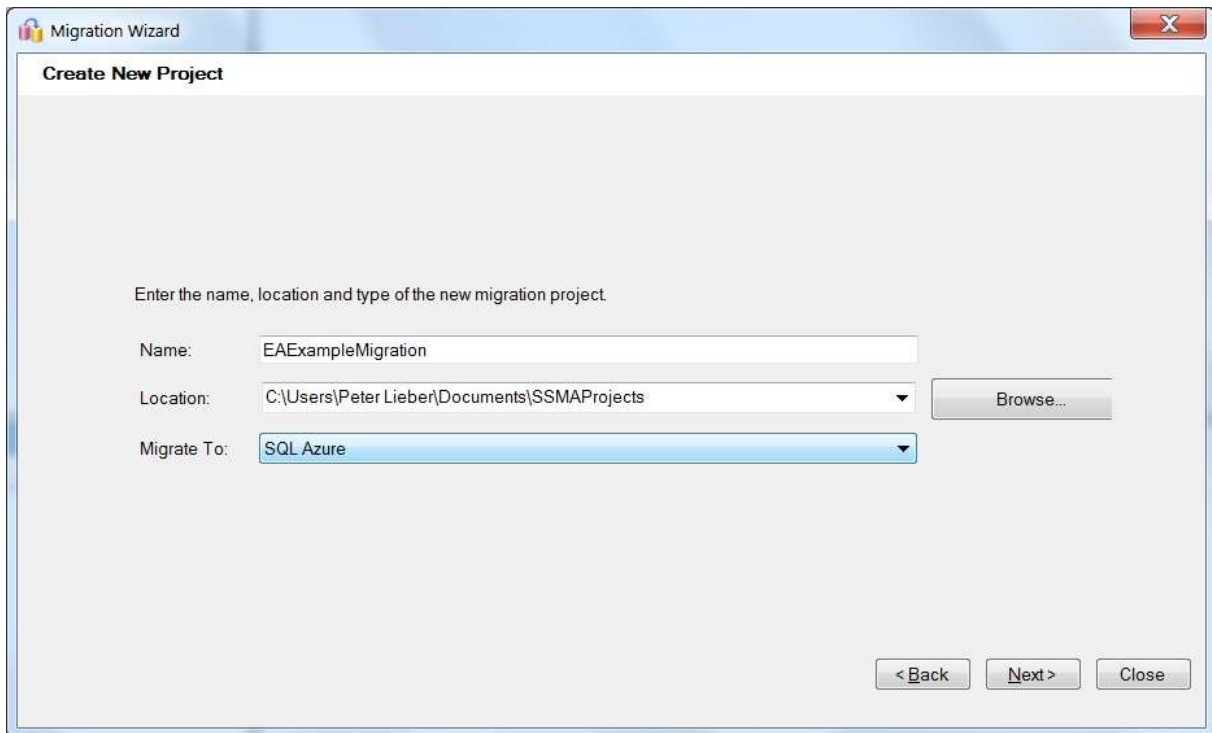
One important rule is to set firewall rules for SQL Azure, but it is allowed to define a range from 0.0.0.0 to 255.255.255.255 and that means available to everybody.

STEP 3: MICROSOFT SQL SERVER MIGRATION ASSISTANT FOR ACCESS (SSMA)

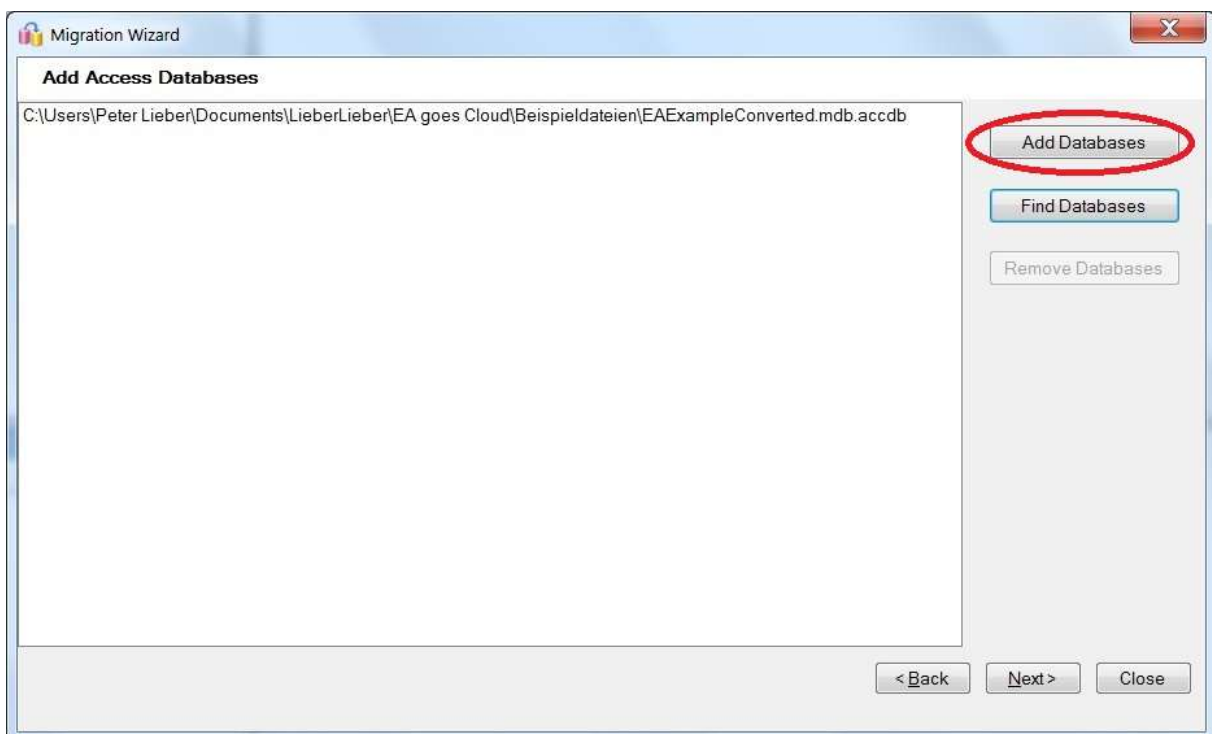
For conversion of an Enterprise Architect Repository the functionality of SSMA Wizard is powerful enough. Just follow the step-by-step wizard of SSMA:



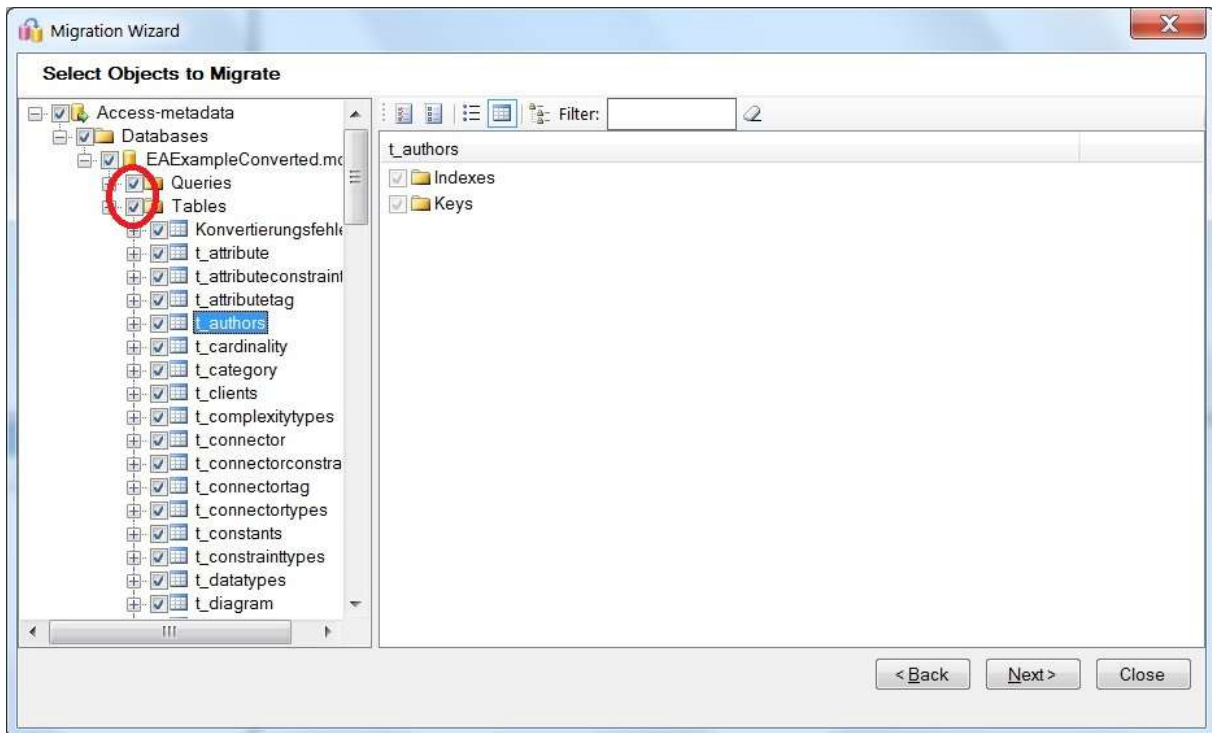
First choose a conversion project name – this makes sense if you want to reproduce conversion with the same repository more often. At Migrate To choose SQL Azure:



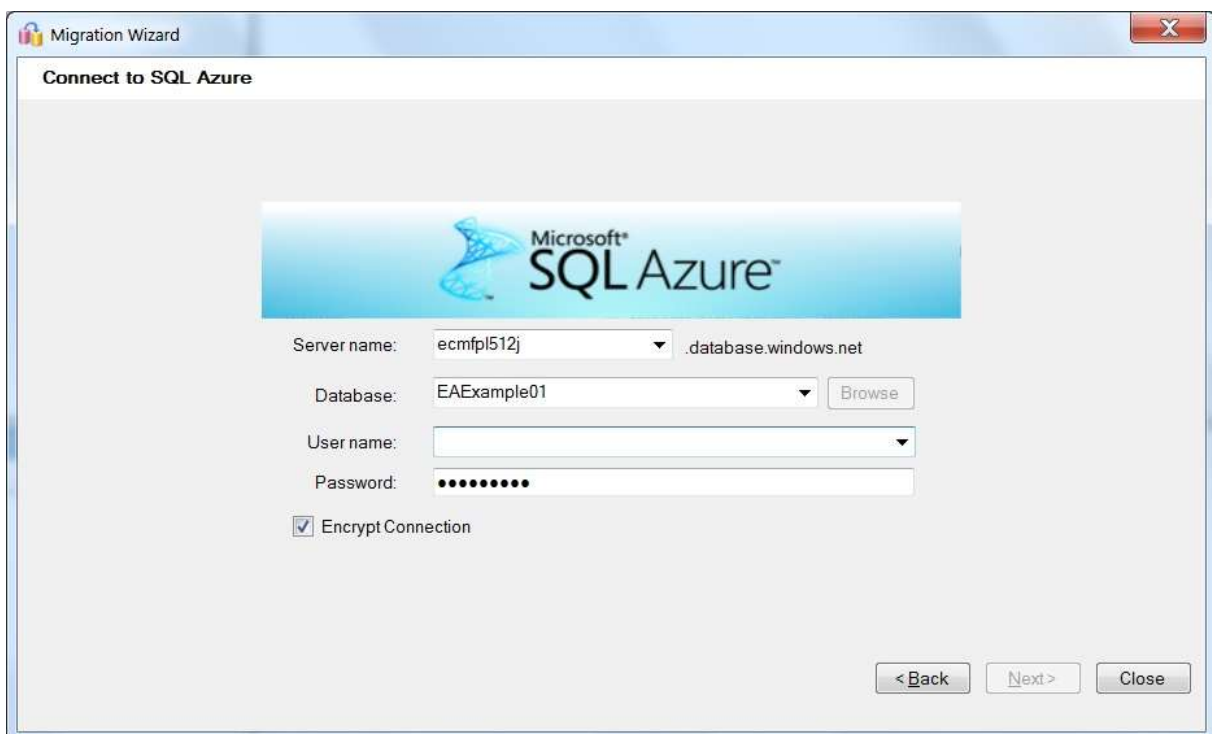
Then choose the converted Access file prepared in step 1:



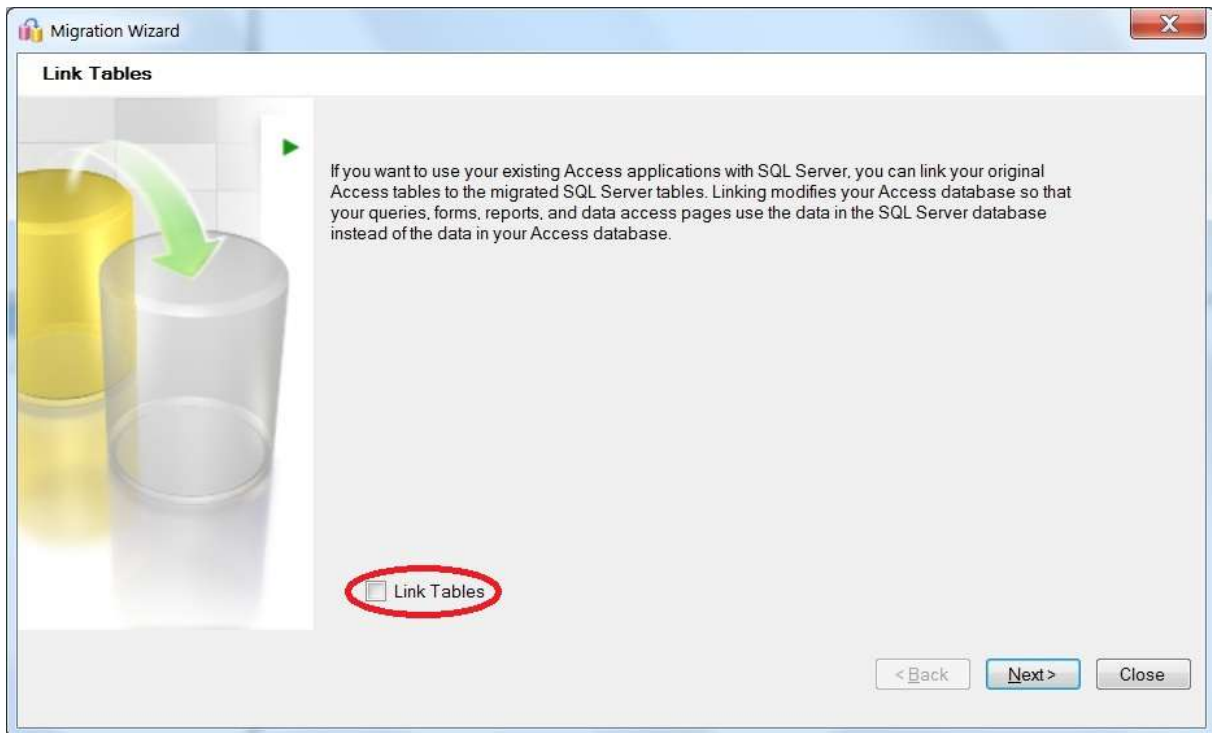
By default only tables are marked – it makes sense also to mark the queries, because they are prepared for some background optimizations of EA and of course esp. required for this sample.



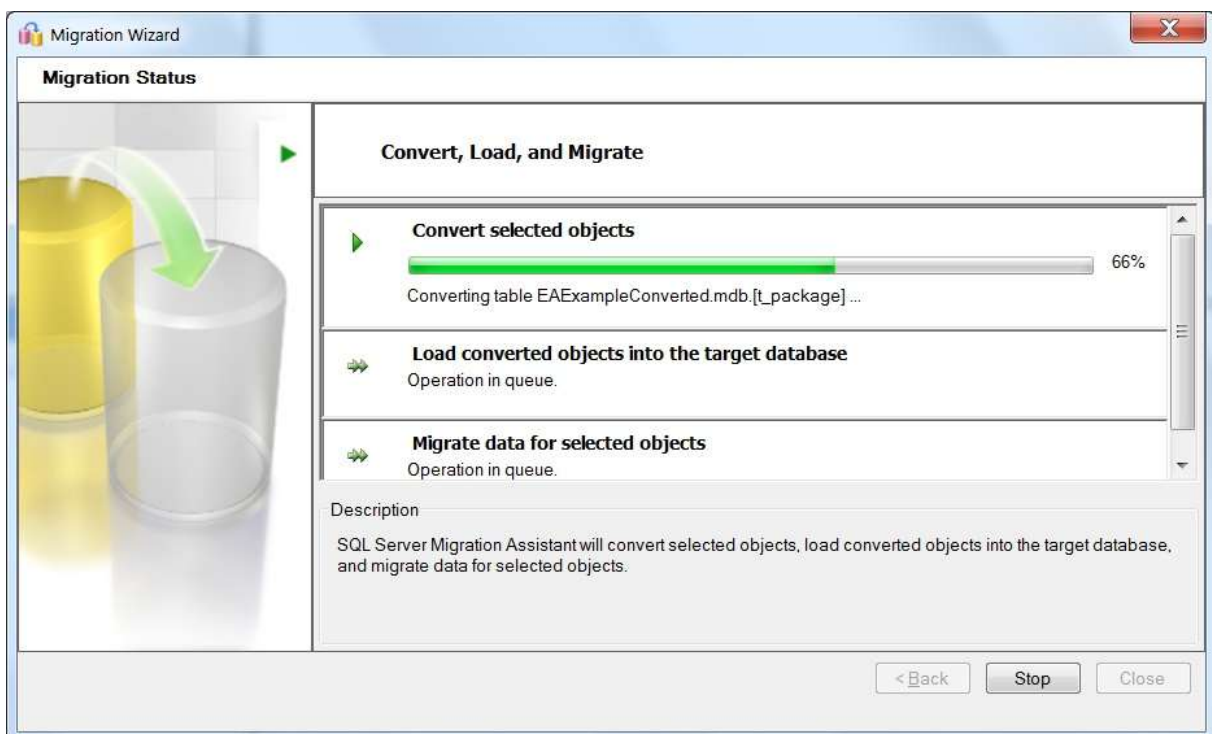
Fill out the SQL Azure connection properties:



You can choose if you want to use the Access file as a proxy for SQL Azure – this is not recommended and this will not work appropriate:



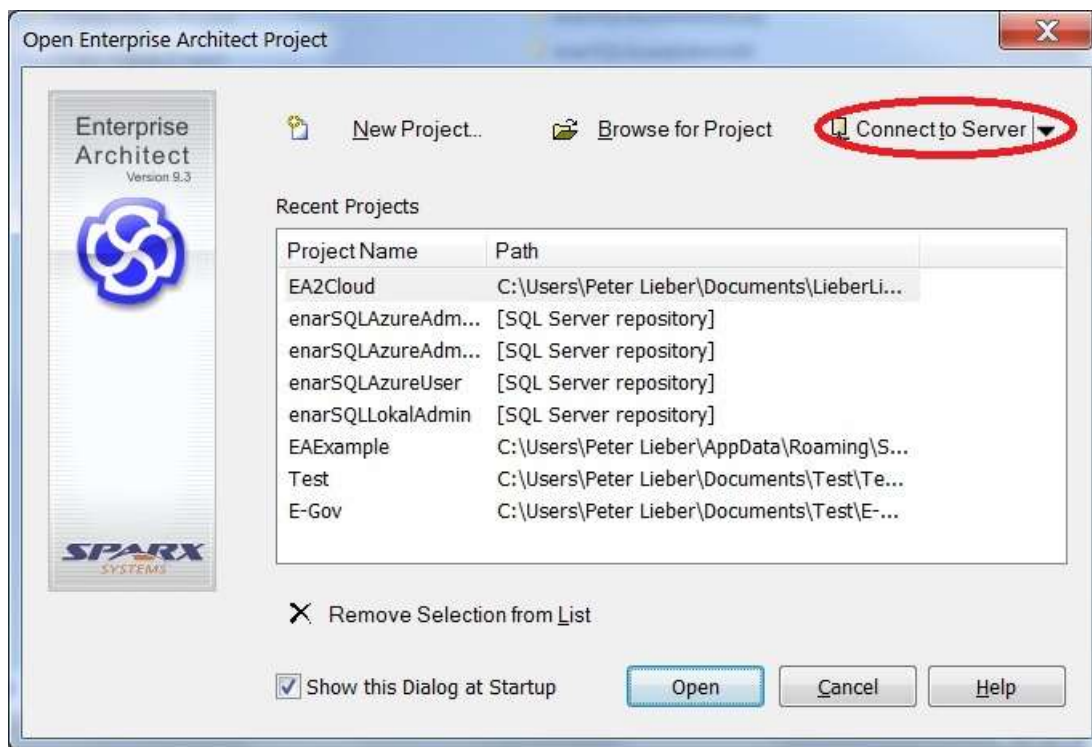
Then the conversion process starts:



Ignore all issues – because they only adress issues in the queries region.

STEP 4: OPEN SQL AZURE BASED REPOSITORY OF ENTERPRISE ARCHITECT

Connect now EA with the new SQL Azure Repository.



~~OLD (VERSION 1.0): CONNECTION USING OLEDB~~

~~OLD (Version 1.0): Use the connection wizard. Choose SQL Server as driver it should work with the native driver to — here we used the OLE DB driver, switch to “Connection” tab and insert details to connect to SQL Azure using the complete name of the database.~~

~~For user use the following format: *EAExampleAdmin@ecmfp1512j*. Check the store the password in connection checkbox.~~

~~The connection string stores now the password so use the encrypted option of EA to make save deployment of the connection string easier. For best performance data chose also LazyLoad. WAN Optimizer is not usable in a SQL Azure environment.~~

NEW (VERSION 2.0) CONNECTION USING SQL NATIVE CLIENT

Do not use the OLE DB Provider, because OLE DB is not supported by SQL Azure – the worse thing it seems to work, but in details EA will have a lot of troubles and will behave strange.

So here comes what you have to do:

- 1.) You have to install SQL Native Client (for ODBC) first – it is part of SQL Server Tools (and also for SQL Server Express Tools, so it is usable for free, afaik).
- 2.) Configure an ODBC DSN – if you are on a 64bit Enviroment take care to configure the DSN using the ODBCAD32.EXE from the “c:\windows\system32\” directory _not_ the one from

the “c:\windows\sysWOW64\” – because the DSNs from both environments will come up within EA (Microsoft says its “by design”).

a. Name the ODBC DSN appropriate – lets say “**EAEexample32**”

3.) Then – within EA configure – using the “Connection String” and yes – you have to type it – because the wizard not really supports everything that is expected. Of course you (and all your users) can see now the password and it is not a good idea to promote the admin password – but for this whitepaper it is fine enough.

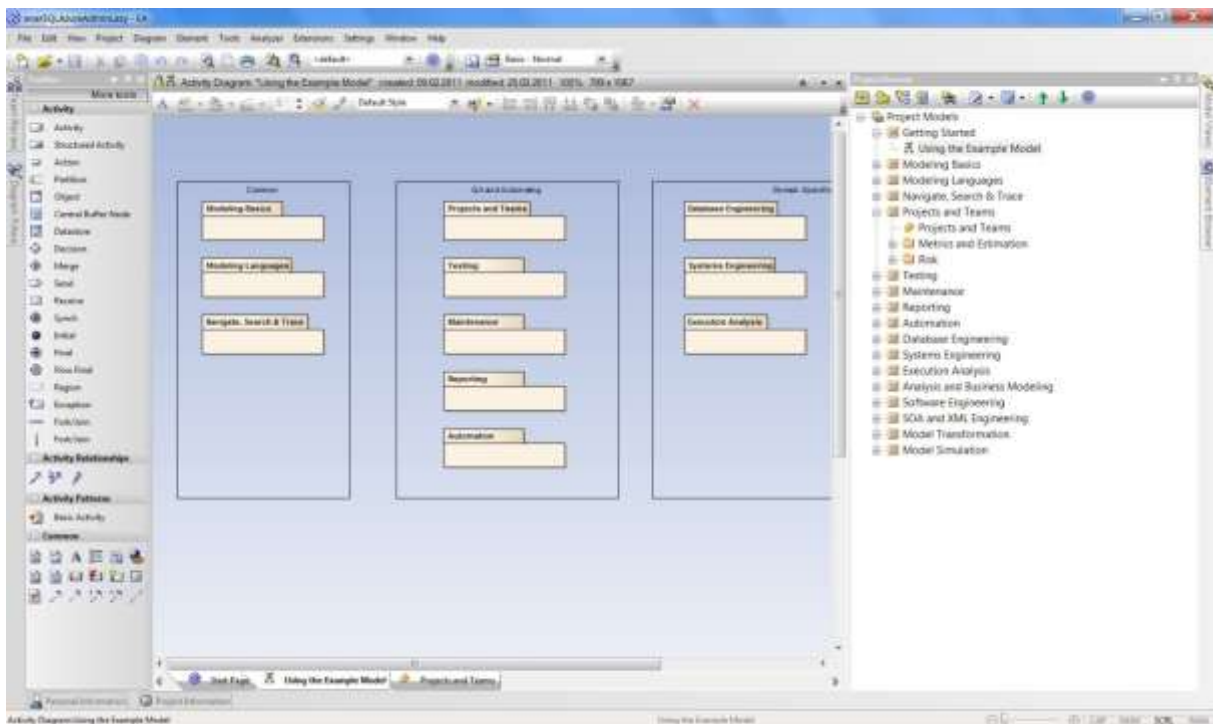
a. EAEexample32 --- DBType=1;Connect=DSN=**EAEexample32**;UID=EAEexampleAdmin@ecmfp1512j.database.windows.net;PWD=topsecret;Initial Catalog=EAEexample01;**LazyLoad=1**;

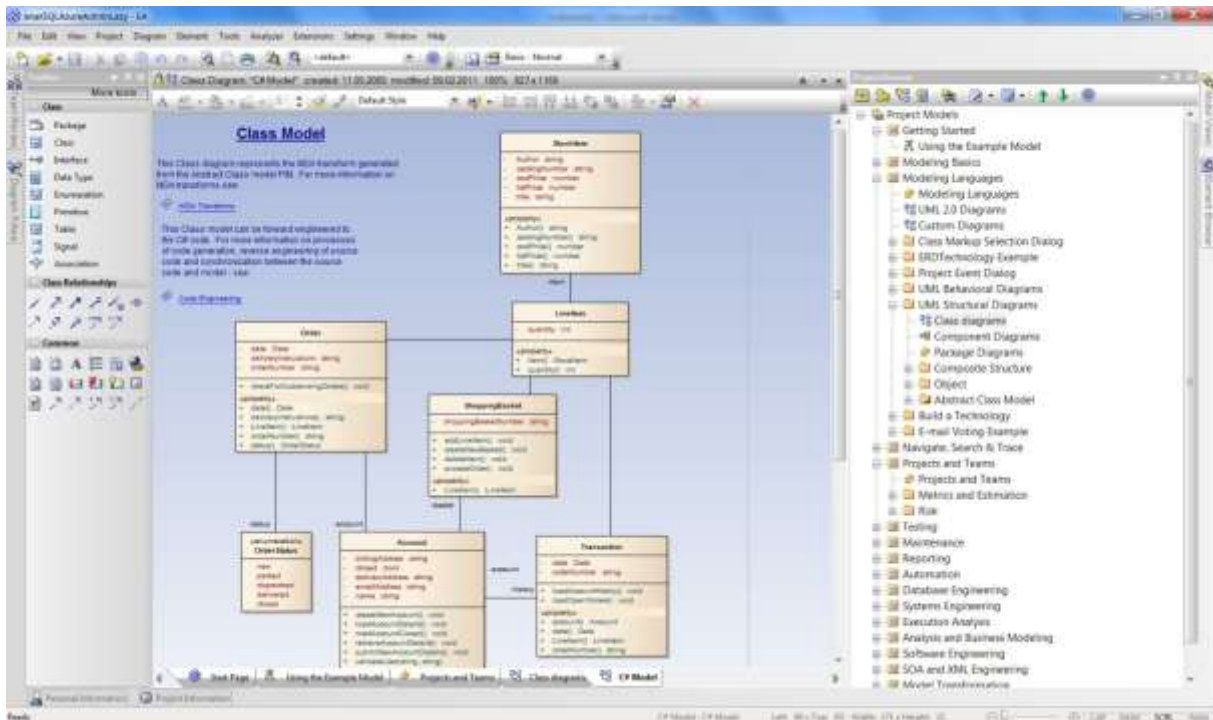
NEW (VERSION 2.0) ATTRACTIVE, ALTERNATIVE SOLUTION: CONNECTION USING CNS MEDIA GATEWAY

If you want to have an attractive, alternative, secure (encrypted) route with a slim ODBC driver on the client side using CORBA and optimized traffic to connect to Microsoft Azure based Server and there connect to the repository then try out CNS Media Gateway. LieberLieber is currently working to provide an Enterprise Architect Plugin for that environment.

RESULTS

Last but not least:





**Welcome Enterprise Architect to the
Microsoft SQL Azure Cloud!**

SOME HINTS

HINT 1: DATABASE CONNECTION

~~OLD: Enterprise Architect recommends a stable database connection, by system architecture this is not guaranteed by SQL Azure and so sometimes (on time out, on changing network, ...) Microsoft OLE DB Provider messages/warnings might occur. Just confirm the messages – after such a message EA tries to reconnect and everything should work fine again.~~

NEW (Version 2.0): if you change the network or you configured a fire wall in SQL Azure – the connection may be lost by EA. EA behaves as EA behaves in this case.

HINT 2: BEING INACTIVE WITH EA

It may happen, that EA asks for a password on longer inactivity – just cancel request (even more than once) – if you are sure that everything should be fine – otherwise try to reopen the project.

HINT 3: FAST INTERNET CONNECTION

For SQL Azure internet connection performance is typically not the issue. But Microsoft recommends a minimum bandwidth of 16Mbit for fluid use. My experience is, that 4Mbit is for EA fast enough for working.

MORE TIPS

TIP 1: PERFORMANCE OPTIMIZATION

It makes sense to insert the following indices to SQL Azure – of course you can run your own optimizations using the tools for SQL Azure.

```
CREATE INDEX [PD1] ON [t_object] ([PDATA1]);
CREATE INDEX [ix_attribute_classifier] ON [t_attribute] ([Classifier]);
CREATE UNIQUE INDEX [ux_attribute_guid] ON [t_attribute] ([ea_guid]);
CREATE INDEX [ix_attribute_type] ON [t_attribute] ([Type]);
CREATE INDEX [ix_attributetag_value] ON [t_attributetag] ([VALUE]);
CREATE INDEX [ix_connector_seqno] ON [t_connector] ([SeqNo]);
CREATE INDEX [ix_connector_eid_connid] ON
[t_connector] ([End_Object_ID], [Connector_ID]);
CREATE INDEX [ix_connector_sid_connid] ON
[t_connector] ([Start_Object_ID], [Connector_ID]);
```

```
CREATE INDEX [ix_connectortag_property] ON [t_connectortag] ([Property]);
CREATE INDEX [ix_connectortag_value] ON [t_connectortag] ([VALUE]);
CREATE INDEX [ix_datatypes_productname] ON [t_datatypes] ([ProductName]);
CREATE INDEX [ix_datatypes_datatype] ON [t_datatypes] ([DataType]);
CREATE INDEX [ix_object_name] ON [t_object] ([Name]);
CREATE INDEX [ix_object_objecttype] ON [t_object] ([Object_Type]);
CREATE INDEX [ix_object_pkgidpdclass] ON
[t_object] ([Package_ID], [PDATA1], [Classifier]);
CREATE INDEX [ix_objectscen_oiidevalscen] ON
[t_objectscenarios] ([Object_ID], [EValue], [Scenario]);
CREATE INDEX [ix_operation_classifier] ON [t_operation] ([Classifier]);
CREATE UNIQUE INDEX [ux_operation_guid] ON [t_operation] ([ea_guid]);
CREATE UNIQUE INDEX [ix_opparams_guid] ON [t_operationparams] ([ea_guid]);
CREATE INDEX [ix_operationntag_value] ON [t_operationntag] ([VALUE]);
CREATE INDEX [ix_t_taggedvalue_value] ON [t_taggedvalue] ([ElementID]);
CREATE INDEX [ix_xrefsystem_client] ON [t_xrefsystem] ([Client]);
CREATE INDEX [ix_xrefsystem_supplier] ON [t_xrefsystem] ([Supplier]);
CREATE INDEX [ix_xrefsystem_type] ON [t_xrefsystem] ([Type]);
CREATE INDEX [ix_xrefuser_client] ON [t_xrefuser] ([Client]);
CREATE INDEX [ix_xrefuser_supplier] ON [t_xrefuser] ([Supplier]);
CREATE INDEX [ix_xrefuser_type] ON [t_xrefuser] ([Type]);
```

TIP 2: WORKING IN TEAMS WITH ENTERPRISE ARCHITECT

Read the recommended whitepaper for deployment of EA for teams:

http://sparxsystems.com/downloads/whitepapers/EA_Deployment.pdf

TIP 3: USE OF REPLICATION AND DATA SYNCHRONISATION

SQL Azure supports an interesting option for interoperability of SQL Servers:

<http://msdn.microsoft.com/en-us/library/hh456371.aspx>

END OF DOCUMENT