

How to transition from the CAST AIP DB2 Analyzer to the SQL Analyzer extension

On this page:

- [Introduction](#)
- [Step 1: Take a snapshot](#)
- [Step 2: Backup CAST AIP schemas and Delivery/Deployment folders](#)
- [Step 3: Check results and grades](#)
- [Step 4: Download and install the latest version of the SQL Analyzer extension](#)
- [Step 5: Retrieve existing DB2 UDB connection details](#)
- [Step 6: Generate SQL source](#)
- [Step 7: Add a new version](#)
- [Step 8: Recalibrate your Application](#)

Target audience:

Users of the **SQL Analyzer** extension.



Summary: this page provides instructions for those that would like to transition from using the "out-of-the box" DB2 Analyzer provided in CAST AIP to the SQL Analyzer extension.

Introduction

If you have been actively analyzing DB2 (z/OS or UDB) with the **DB2 Analyzer** (provided out-of-the-box in CAST AIP) you can transition to using the SQL Analyzer extension to analyze your DB2 source code. The process of transitioning is described in this page.



Note that it is only possible to analyze DB2 when using the SQL Analyzer extension 2.x.

Step 1: Take a snapshot

Take a snapshot of the existing Application prior to making any further changes.

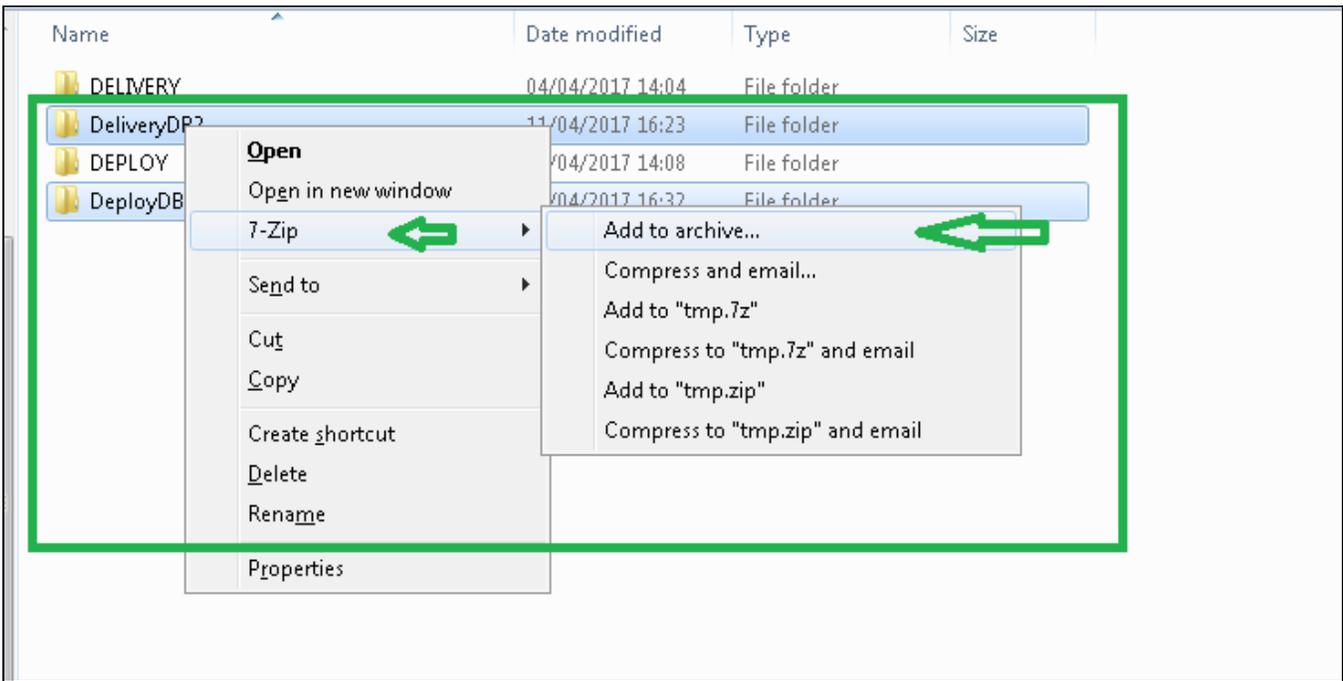
Step 2: Backup CAST AIP schemas and Delivery/Deployment folders

- To backup the CAST AIP schemas use the [CSSBackup tool](#) (located in the **CSSAdmin** folder in the CAST AIP installation folder) use the following scripts adapted to your environment:

Backup CAST AIP schemas

```
CSSBackup.exe -schema db2_udb_local -password CastAIP -file c:\temp\DB2\db2_udb_local.cssdmp -log c:\temp\DB2\db2_udb_local.log
CSSBackup.exe -schema db2_udb_central -password CastAIP -file c:\temp\DB2\db2_udb_central.cssdmp -log c:\temp\DB2\db2_udb_central.log
CSSBackup.exe -schema db2_udb_measure -password CastAIP -file c:\temp\DB2\db2_udb_measure.cssdmp -log c:\temp\DB2\db2_udb_measure.log
CSSBackup.exe -schema db2_udb_mngt -password CastAIP -file c:\temp\DB2\db2_udb_mngt.cssdmp -log c:\temp\DB2\db2_udb_mngt.log
```

- Then ZIP the **Delivery** and **Deployment folders** for easy restoration:

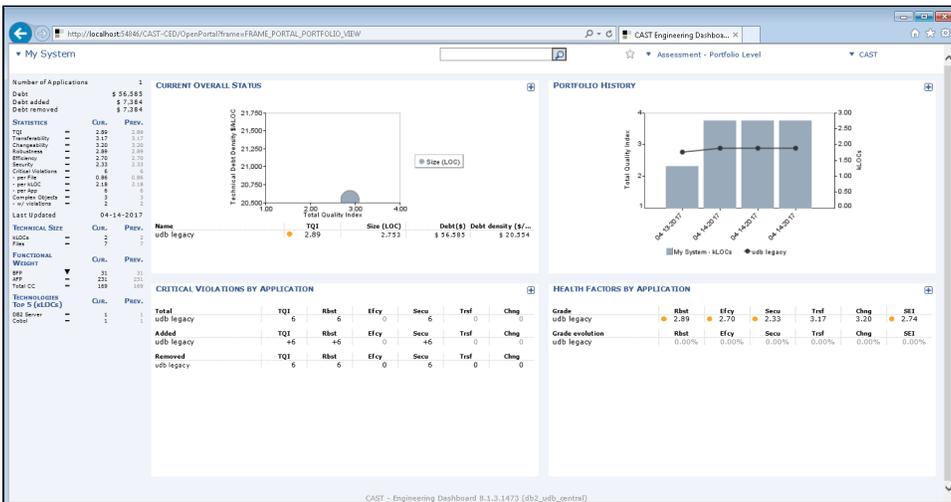


Step 3: Check results and grades

Before proceeding, check your results and grades using the CAST dashboards. Example below show a DB2 UDB analysis:

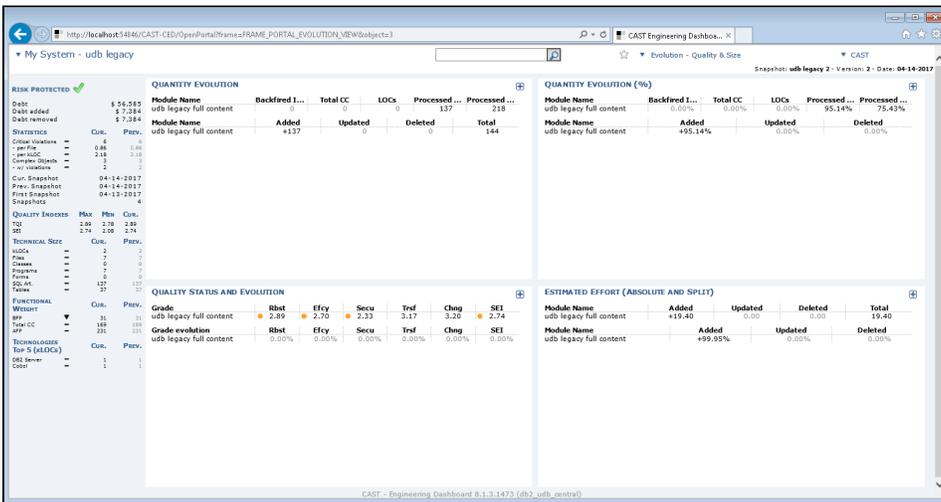
Portfolio

Click to enlarge:



Quality and size evolution

Click to enlarge:



Violated rules by technology

Click to enlarge:

LIST OF VIOLATED RULES BY TECHNOLOGIES	
Rule Name	Nb of Violation
Avoid undocumented Triggers, Functions and Procedures	97
Avoid triggers, functions and procedures with a very low comment/code ratio	97
Avoid Artifacts with SQL statement including subqueries	44
Avoid Artifacts with lines longer than X characters	40
Never use SQL queries with a cartesian product	32
View naming convention - name size control	26
Avoid Tables not using referential integrity	25
Avoid unreferenced views	22
Table naming convention - name size control	22
Avoid SQL queries not using the first column of a composite index in the WHERE clause	21
Avoid using Cursors	14
Avoid SQL queries with implicit conversions in the WHERE clause	12
Avoid "SELECT *" queries	8
Avoid Artifacts with a Complex SELECT Clause	7
Avoid SQL queries using functions on indexed columns in the WHERE clause	7
Avoid using GOTO statement	4
Avoid Artifacts with Group By	3
Avoid Too Many Copy Pasted Artifacts	3
Avoid Artifacts with High Fan-In	2
Avoid having multiple Artifacts inserting data on the same SQL Table	2
Avoid unreferenced Tables	2
Avoid large Artifacts - too many Lines of Code	2
Avoid Tables with more than 20 columns on an OLTP system	1
Avoid having multiple artifacts deleting data on the same SQL table	1

Step 4: Download and install the latest version of the SQL Analyzer extension

Please see:

- <http://doc.castsoftware.com/display/EXTEND/Download+an+extension>
- <http://doc.castsoftware.com/display/EXTEND/Install+an+extension>



Note that you need to install the extension into the CAST AIP schemas that you were using previously to host your Application containing the DB2 analysis results.

Step 5: Retrieve existing DB2 UDB connection details



This step is **only** relevant for those analyzing DB2 UDB online.

Use the following query, run against your CAST AIP Management Service schema, to retrieve the connection details to the live DB2 UDB databases that are used during the analysis with the **DB2 Analyzer** (provided out-of-the-box in CAST AIP). This information can be reused when extracting the DDL for use with the SQL Analyzer extension:

For Oracle and CSS (postgreSQL) Management databases::

```
select partdetails.Object_Name PackageName, Username, Host, Port, DbName, projdetails.Object_Name SchemaName
from
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DatabaseServerParticipatingDB2') part,
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DB2UDBSourceCodeRepository') proj,
CMS_INF_PARTP_DBUDB partdetails,
CMS_UDB_Project projdetails
where part.Field_Value like proj.Field_Value || '%'
and partdetails.Object_ID = part.Object_ID
and proj.Object_ID = projdetails.Resource_ID
order by Host, DbName, SchemaName
```

For Microsoft SQL Management databases:

```
select partdetails.Object_Name PackageName, Username, Host, Port, DbName, projdetails.Object_Name SchemaName
from
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DatabaseServerParticipatingDB2') part,
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DB2UDBSourceCodeRepository') proj,
CMS_INF_PARTP_DBUDB partdetails,
CMS_UDB_Project projdetails
where part.Field_Value like proj.Field_Value + '%'
and partdetails.Object_ID = part.Object_ID
and proj.Object_ID = projdetails.Resource_ID
order by Host, DbName, SchemaName
```

The output of the above queries will result in similar output

CAST DMT Package Name	DB2 login	Host name	Host port	Database Name	Schema Name
UDB_Package	db2admin	UDBSERVER	50000	TEST	CASTPUBS

Step 6: Generate SQL source

This step explains how to generate the DDL from the DB2 database that will be fed into the **CAST Delivery Manager Tool** for analysis with the SQL Analyzer extension.



- See also [SQL Analyzer - generate DDL for a DB2 database](#) for more information.
- Note that the **db2look tool** is only available with an installation of a **DB2 UDB instance**. If you are using a **DB2 z/OS instance**, CAST recommends the use of the **RC/Migrator** tool which is part of the **CA Database Management Solution for DB2 for Z/OS, Version 19**. There are no instructions provided for this.

db2look command line

The tool used to achieve this is the **db2look tool** provided with the DB2 UDB server (see the full documentation for the tool here: https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.5.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0002051.html). This example will generate an .SQL file containing the DDL for the CASTPUBS schema on the TEST database accessed with the db2admin/db2admin credentials:

Generate db2look commands:

```
db2look -d TEST -z CASTPUBS -i db2admin -w db2admin -e -o E:\tmp\SQL_DB2\TEST_CASTPUBS.sql
```

You can view **example DDL** output generated by db2look in the attached files:

- [CASTPUBS](#)
- [DB2_DIAGS](#)

Generate db2look commands using a query



This section is **only** relevant for those analyzing DB2 UDB online.

If you need to generate DDL for **multiple schemas** use the following query (run against your CAST AIP Management Service schema) to automatically generate commands that can be used with the **db2look** tool:

For Oracle and CSS (postgreSQL) Management databases:

```
select 'db2look -d ' || DbName || ' -z ' || projdetails.Object_Name || ' -i ' || UserName || ' -w db2admin -e -o ' || DbName || '_' || projdetails.Object_Name || '_DDL.sql' batch
from
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DatabaseServerParticipatingDB2') part,
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DB2UDBSourceCodeRepository') proj,
CMS_INF_PARTP_DBUDB partdetails,
CMS_UDB_Project projdetails
where part.Field_Value like proj.Field_Value || '%'
and partdetails.Object_ID = part.Object_ID
and proj.Object_ID = projdetails.Resource_ID
order by batch
```

For Microsoft SQL Management databases:

```
select 'db2look -d ' + DbName + ' -z ' + projdetails.Object_Name + ' -i ' + UserName + ' -w db2admin -e -o ' + DbName + '_' + projdetails.Object_Name + '_DDL.sql' batch
from
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DatabaseServerParticipatingDB2') part,
(select * from CMS_DynamicFields where Entity_GUID = 'dbtwoserver.DB2UDBSourceCodeRepository') proj,
CMS_INF_PARTP_DBUDB partdetails,
CMS_UDB_Project projdetails
where part.Field_Value like proj.Field_Value + '%'
and partdetails.Object_ID = part.Object_ID
and proj.Object_ID = projdetails.Resource_ID
order by batch
```

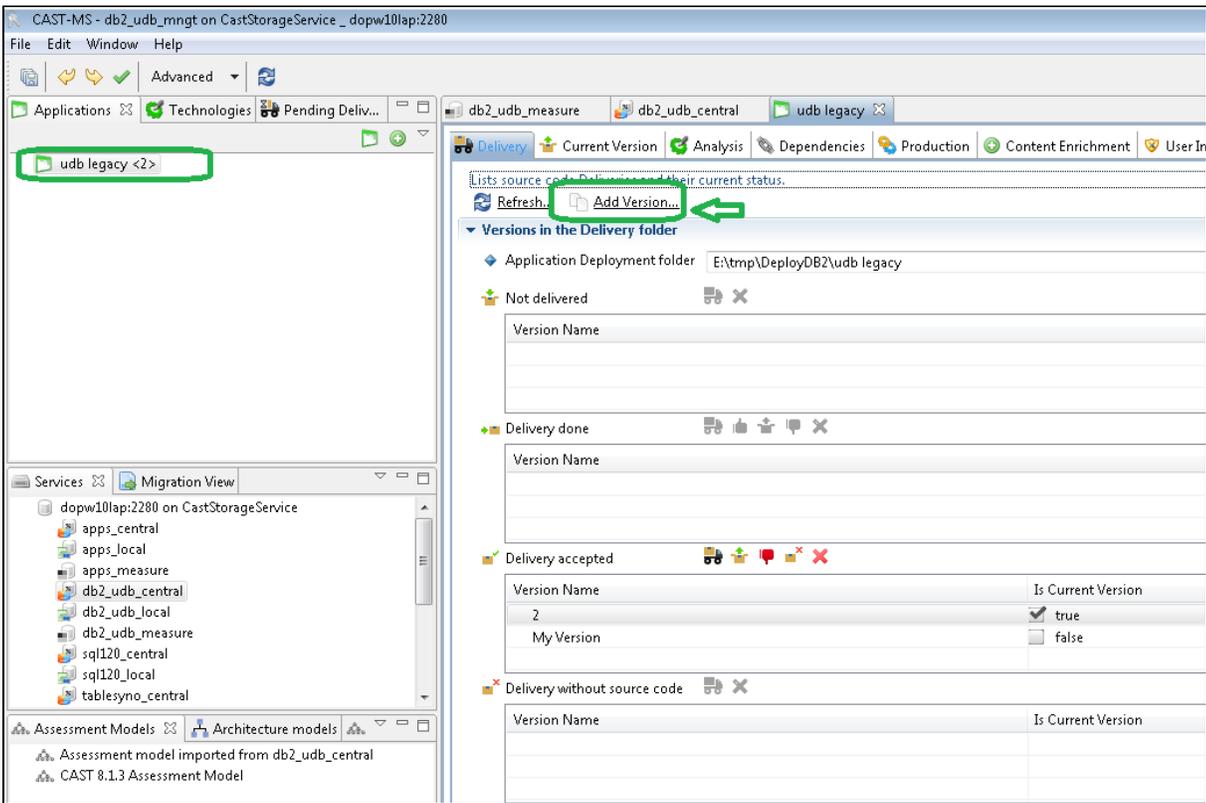
The result of the above queries will be as follows - **one db2look command per schema** on the target DB2 server. Each command will output the specified schema as DDL into one .SQL file:

Resulting db2look commands

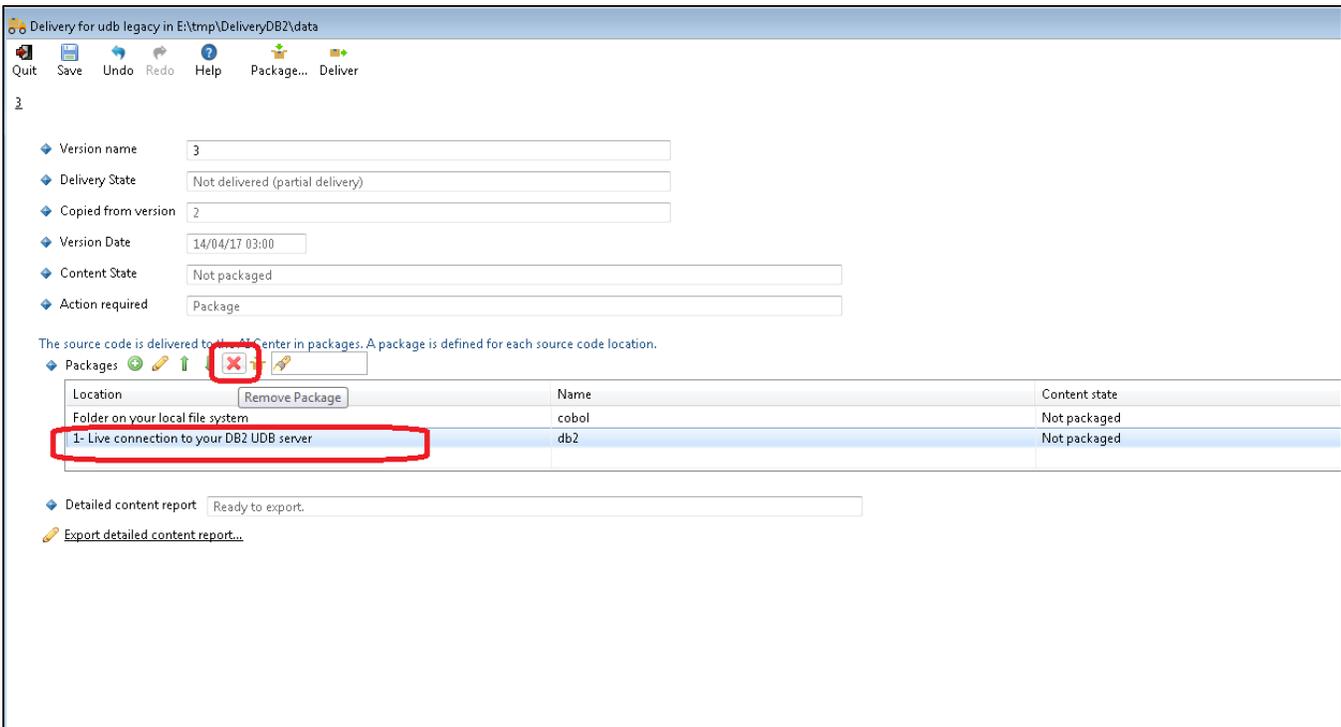
```
db2look -d TEST -z CASTPUBS -i db2admin -w db2admin -e -o E:\tmp\SQL_DB2\CASTPUBS.sql
db2look -d TEST -z DB2_DIAGS -i db2admin -w db2admin -e -o E:\tmp\SQL_DB2\DB2_DIAGS.sql
```

Step 7: Add a new version

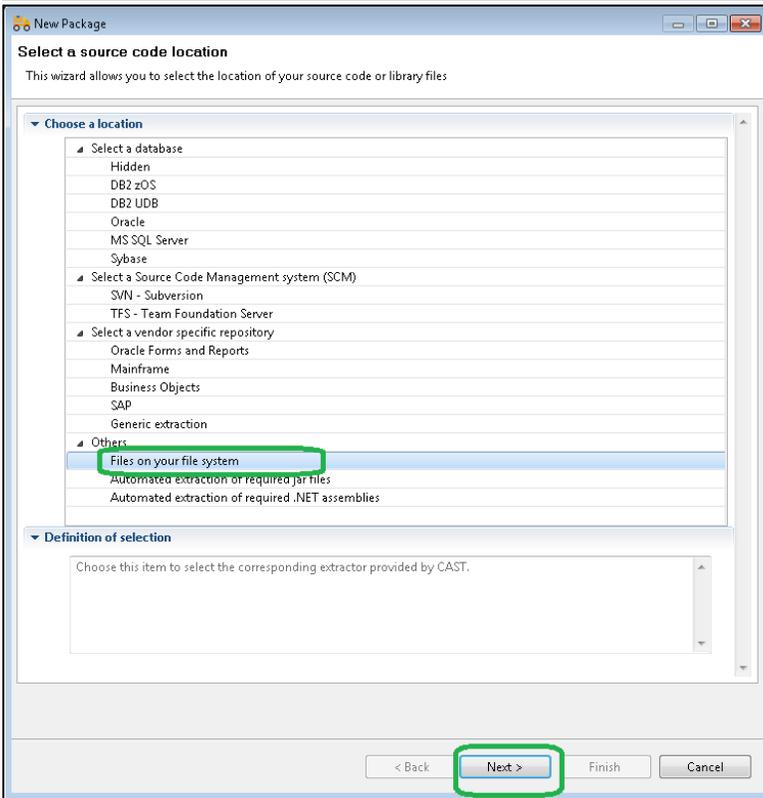
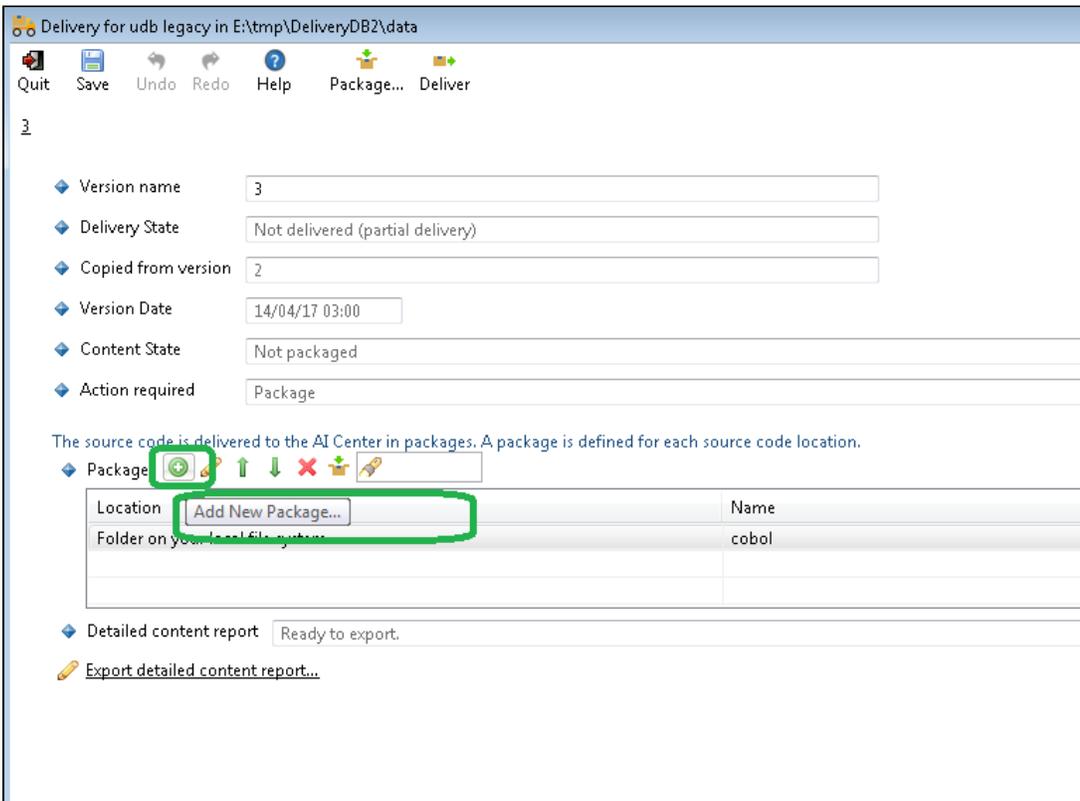
- In the CAST Management Studio, add a new version for the existing Application in which the online DB2 UDB analysis or the z/OS analysis is configured. Examples below show "DB2 UDB online" (*click to enlarge*):



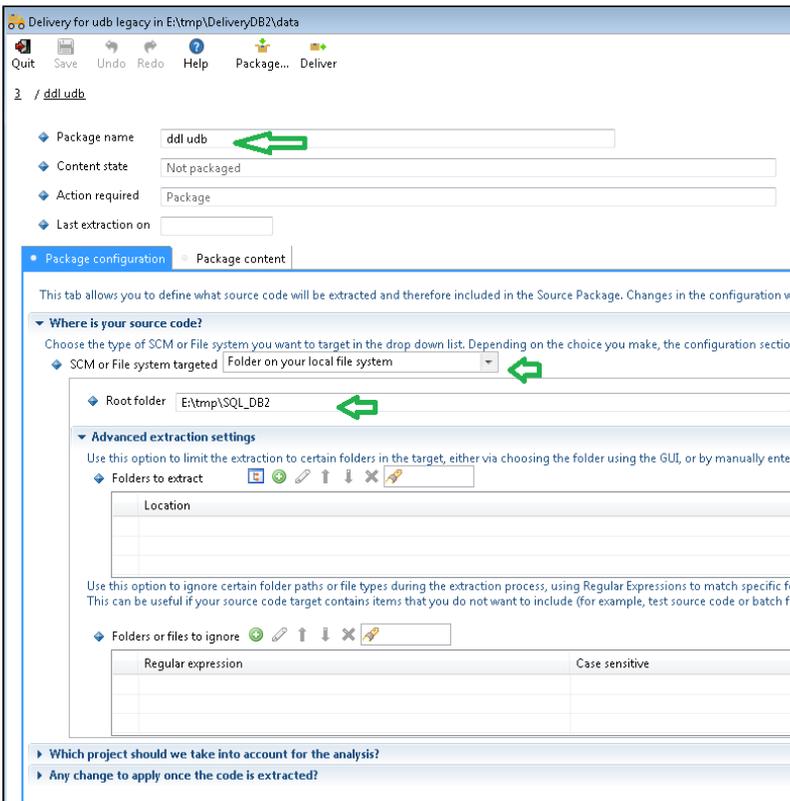
- In the CAST Delivery Manager Tool, remove the existing DB2 UDB online or z/OS package as shown below (*click to enlarge*):



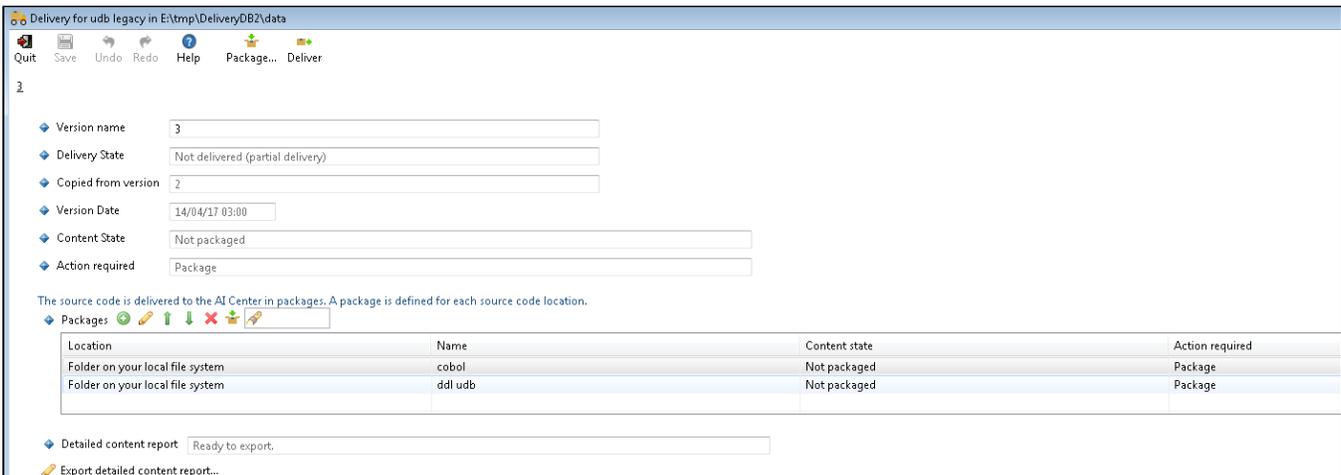
- Add a new package to deliver the DB2 DDL that you have already generated, using the **Files on your system** option (*click to enlarge*):



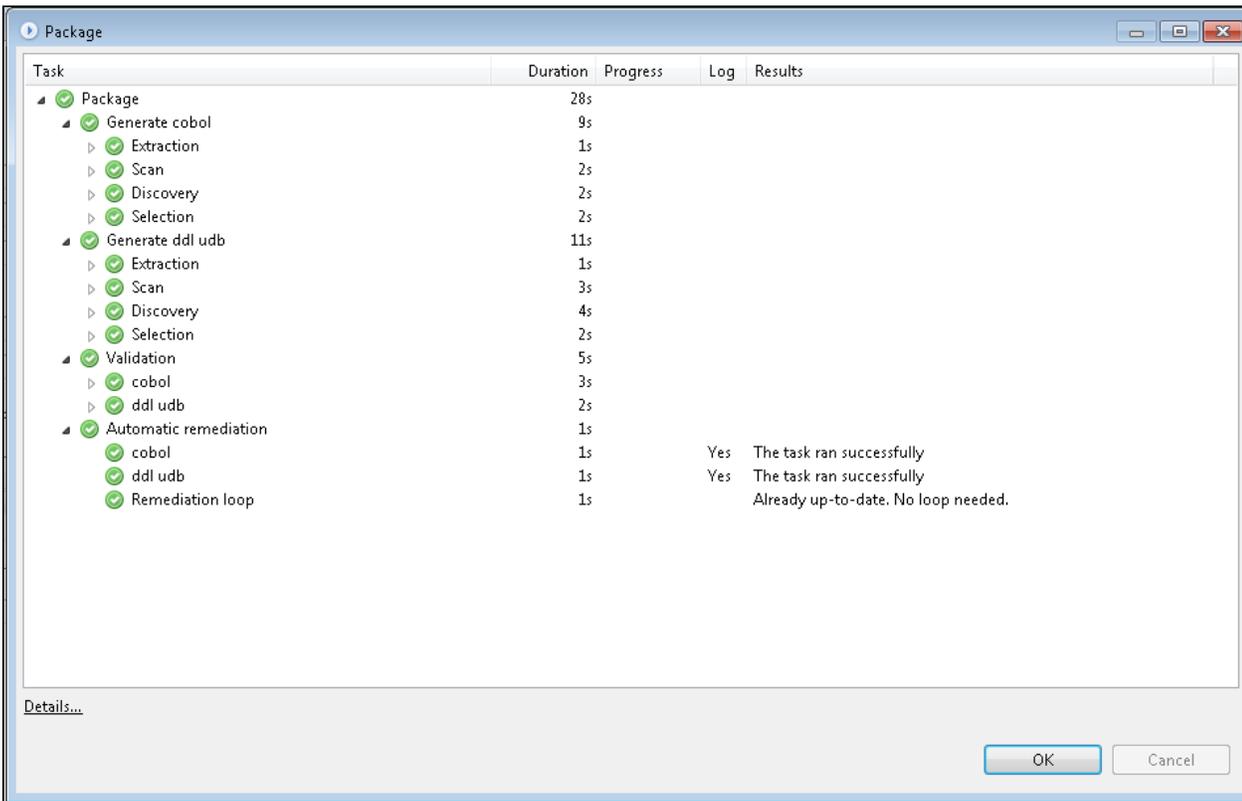
- Configure the package and **define the path** to the folder that contains the .SQL files generated by the **db2look tool** (*click to enlarge*):



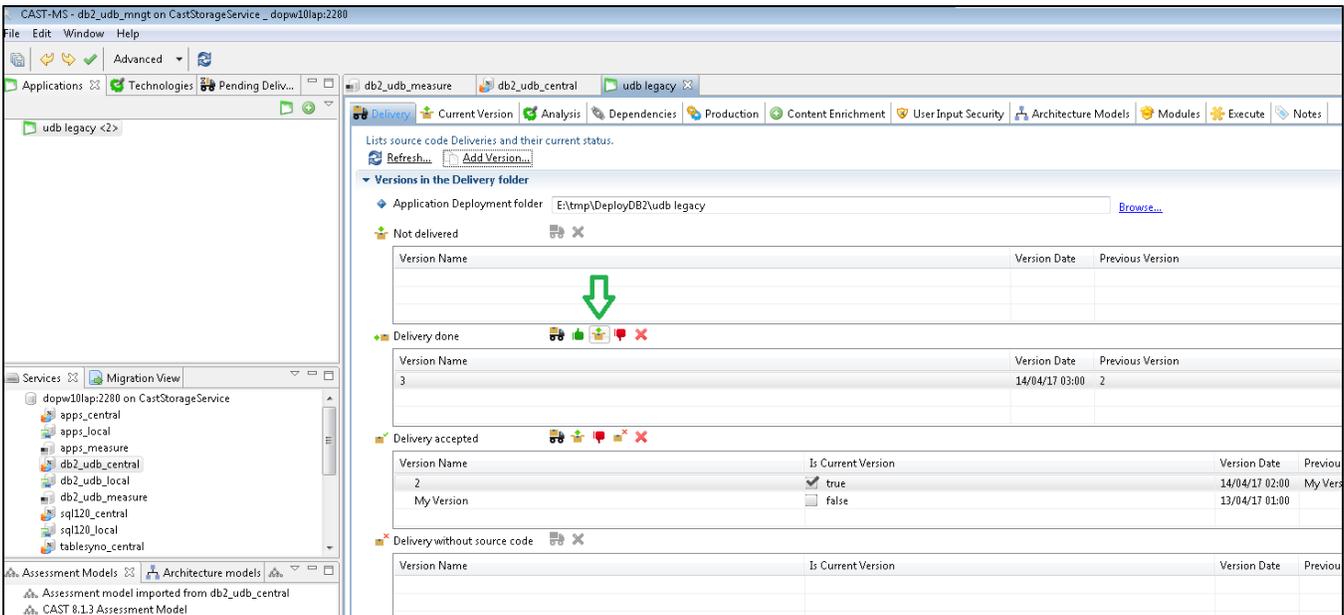
- As shown below, the new package called **ddl udb** is listed along side the existing "cobol" package (*click to enlarge*):



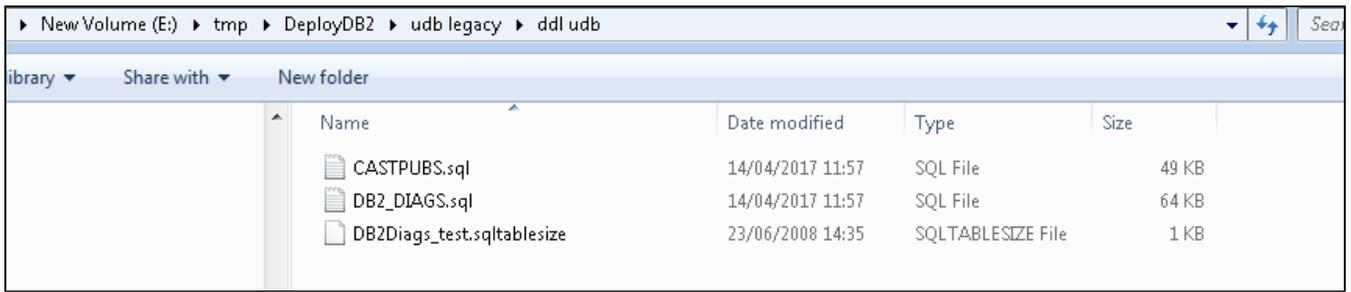
- Run the packaging action (*click to enlarge*):



- Deliver and then set as the current version in the CAST Management Studio (*click to enlarge*):



- Once the "set as current version" action is complete, the delivered .SQL files can be seen in the **Deployment** folder:



- Add a new **Universal Analyzer Analysis Unit** pointing to the .SQL files in the Deployment folder, ensuring the **SQL Analyzer language** is ticked (*click to enlarge*):

Delivery | Current Version | Analysis | Dependencies | Production | Content Enrichment | User Input Security | Architecture Models

This section lists the current Version of the Application's source code and all the packages that have been deployed for analysis.

3

Deployed Packages

Type	Package name
File System Package	cobol
File System Package	ddl udb

Lists the Analysis Units for the Package selected above.

Analysis units

Name	Subject Path	Analyze

Deployment Folder:

Deployment Folder:

Deployment Folder:

Clean up Analysis Units

Name: ddl udb

Analysis Unit description

Source Settings | Production | Execute

Project Path: User defined

Universal language

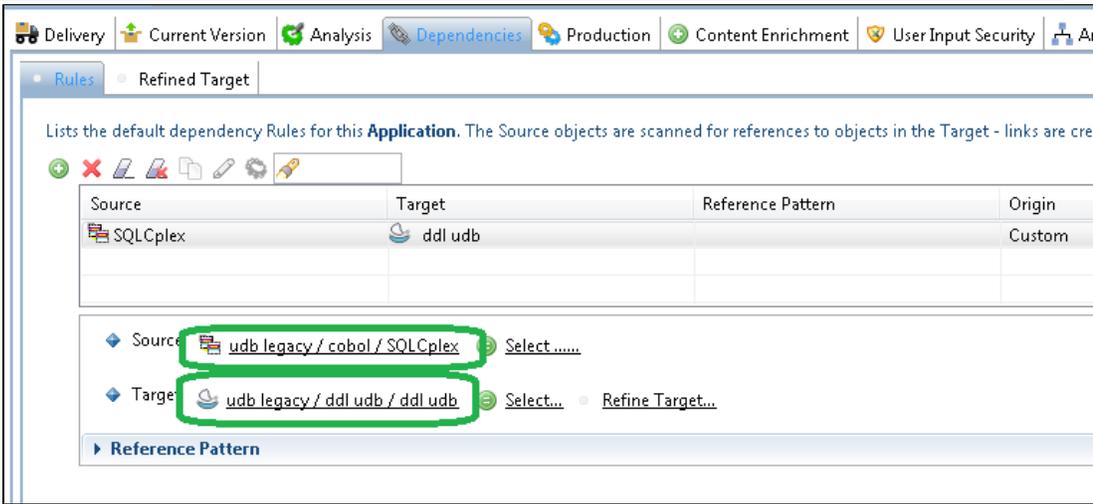
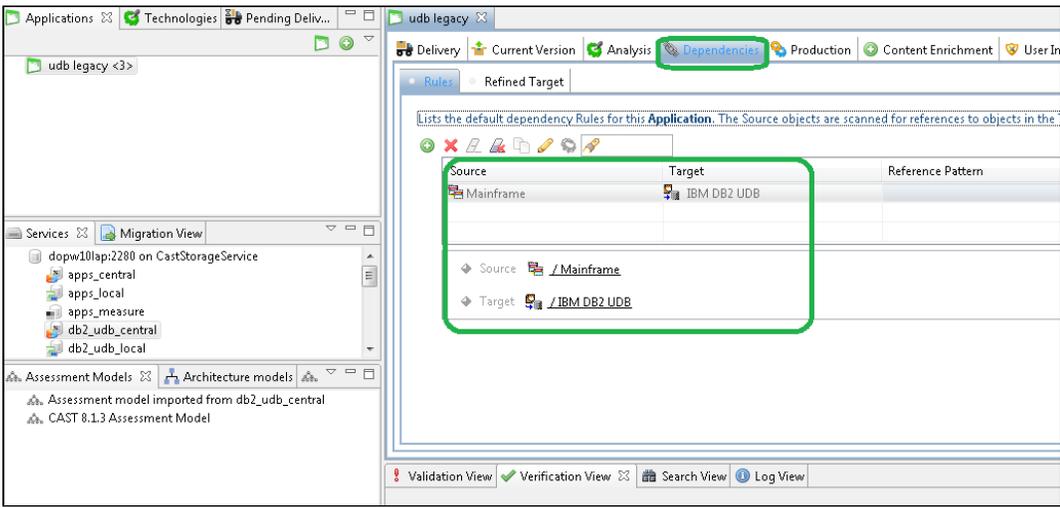
Description

SQL Analyzer

Sources

Permission	Path
Include	E:\tmp\DeployDB2\udb legacy\ddl udb

- Modify the dependencies if you necessary. In this example we have a Mainframe Cobol Analysis Unit and therefore we will replace the existing **Mainframe / IBM DB2 UDB auto created dependency** with **custom cobol / ddl udb dependency** (click to enlarge):



Step 8: Recalibrate your Application

Take **two new snapshots** (ensure that you run an analysis as well) and then check the results in the CAST dashboards:

Portfolio

Click to enlarge:



Quality and size evolution

Click to enlarge:

SQL Analyzer - udb legacy

LIST OF VIOLATED RULES BY TECHNOLOGIES

Rule Name	Nb of Violation
Avoid unreferenced Functions	94
Avoid Artifacts with SQL statement including subqueries	39
Avoid Tables without Primary Key	33
Avoid Tables not using referential integrity	25
Avoid unreferenced views	22
Avoid undocumented Triggers, Functions and Procedures	15
Avoid "SELECT *" queries	10
Avoid using the GROUP BY clause	3
Avoid unreferenced Tables	2
Avoid having multiple Artifacts inserting data on the same SQL Table	2
Avoid Artifacts with High Fan-In	2
Avoid redundant indexes	1
Avoid having multiple artifacts deleting data on the same SQL table	1



Note that new Quality Rules will be calculated for the SQL Analyzer extension in comparison to the online DB2 analyzer:

- Avoid unreferenced Functions
- Avoid redundant indexes
- Avoid Tables without Primary Key