



Hello Readers

As we travel around our respective countries, we see and hear that more and more clients are moving towards testing / installing R8.x. As they do that, they realize that one of the key items moving forward is Training.

Clients that have taken our ERwin R8 one day upgrade training class with hands on exercise's say it is the simplest way to get the team up to speed.

We continue to offer this class like this to our customers. Please contact your local Sandhill contact for more information.

In addition, we would also recommend the following publication for R8 users. "Data Modeling Made Simple - With CA ERwin Data Modeler R8". This is a very well thought out walk through of some of the key features of ERwin R8.x

In additional to all the great updates coming this year, Sandhill will have an additional announcement around a key value add offering on Standards that we believe will be of interest to our clients around the world. (Stay tuned for more information)

As always the latest versions of ERwin are: V7.3.11 and V8.1 If you would like a copy of these build's and you are on maintenance, please let us know.

Regards

Robert Lutton,
Robert.lutton@sandhillconsultants.com
Editor, ERwin Times



Design Layers have changed in ERwin R8.

By Steve Dillon

For those of you who already use design layers, you might want to take note. For those who have never needed to use them in the past, perhaps now is the time to start.

First of all, let me start by explaining what design layers are.

Design layers were introduced in version 4.0 of ERwin Data Modeler, to allow you to create separate logical, physical and combined (classic) models while also allowing you to keep them synchronized. As well as giving you the option of creating stand-alone logical-only and physical-only models, v4 introduced capabilities to split, link and derive models, creating a "map" between objects in the linked models and to be able to synchronize changes across layers using this map.

Another important part of the Design Layer approach was the facility to create Transforms. A wizard allowed you to perform certain common design transformations, such as many-to-many resolution, generalization hierarchy implementation and column demoralization, and keep a record of what the source and target objects were, in a new "Transform" object.

Version 7 introduced extra transform functionality, including the ability to see the source objects in the logical view and target objects in the physical view of a combined model.

The Transform object does not exist in r8.

The transformation wizard still lets you transform the same set of structures, but the transform object with its mapping between source and target objects is not created. This means that the features that relied on this object, such as reversing the transform, or showing different states in the logical and physical views, are no longer available.

You need to decide what to do with your existing r7 models if they contain transforms. You will be faced with a decision when opening the model in r8 as to what state you want to see them in, either source or target, or whatever the current state happens to be.

For combined models, my recommendation is that you upgrade the model twice: once preserving the source state; and then again preserving the target state. I would then discard the physical side of the model in source state and use this as your stand-alone logical model (using split L/P). I would, however, retain both parts of the target-state model. You can then add the stand-alone logical model as a model source to the target model. Unfortunately, you will not be able to fully map transform source objects.

The reason for retaining both logical and physical sides of the target model (and this is something that those already using separate logical and physical models should take note of), is that you now get no choice in how you transform logical-only structures (i.e. those that cannot exist in the physical model, namely m:n relationships and generalization hierarchies) when deriving a physical model from a logical one. The default transformation is applied, and there is no way of changing this in the physical model, as no source objects are visible to transform.

If you derive a combined logical/physical model from a logical model, you can decide how to transform these structures in the logical side of the model. This is still, however, a once-only change.

In short, then, if you currently rely on a single combined model to show both source and target objects of a transform, consider using separate models to do this at r8. If you currently do this, but use physical-only models as your target, consider making your target models combined logical/physical.

Reporting Options in V7 vs R8 – By Ed Bantegui

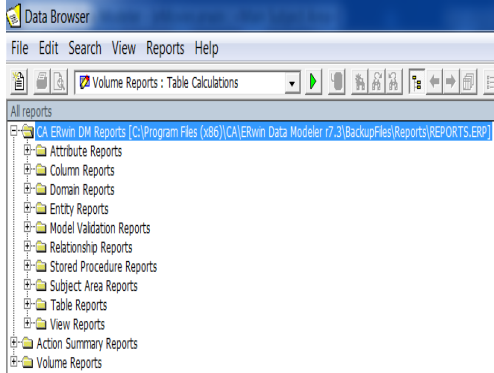
As we travel around N.A., speaking with clients we often hear two items that clients mostly want help with. The first is Complete Compare, and the second is the reporting and output functionality in R8. In regards to CC we have a one day class that covers everything you would want to know about CC and beyond.

In regards to reporting and its output, we have provided some guidance on the functionality used in R7 and where to find this in R8. This article is to show that there are very good reporting features in R8 that users can use. Sandhill has developed a one day reporting class (covering more information in detail, including the Meta model and ODBC Access) for Erwin R8. If anyone would like more information on this, please let us know

ERwin R7 Data Functionality

Data Browser - Figure 1

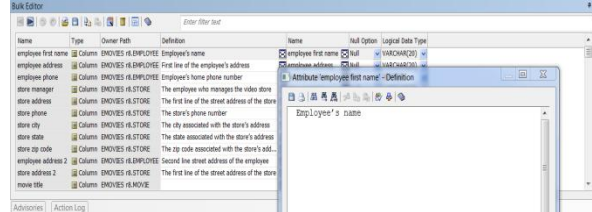
Select /Edit information in the model Singularly



ERwin R8 Functionality

Bulk Editor - Figure 2

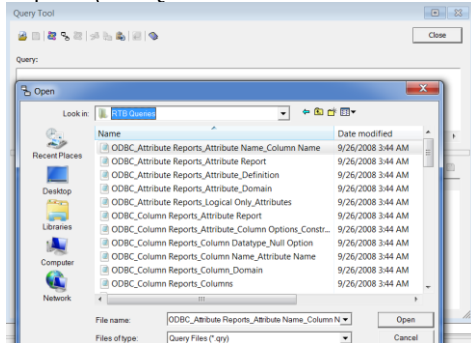
Select / Edit Multiple Objects using Wizard



Query Based Reporting - Figure 3

Reports found in

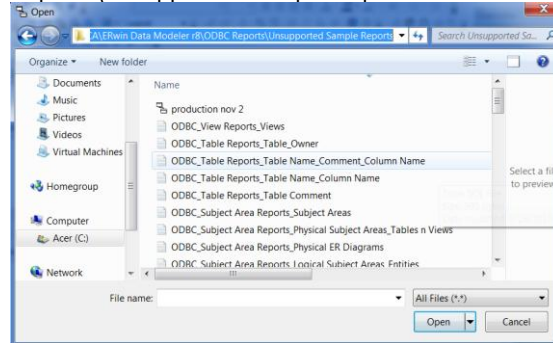
C:\Program Files (x86)\CA\ERwin Data Modeler r7.3\ODBC Reports\RTB Queries



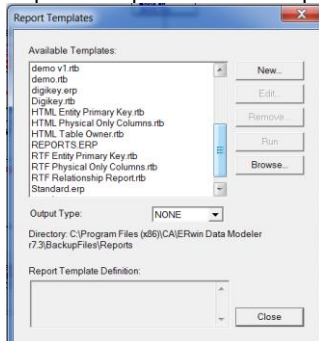
Query Based Reporting - Figure 4

Reports found in File type is "*.SQL"

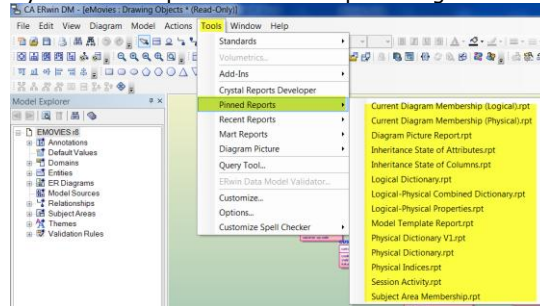
C:\Program Files (x86)\CA\ERwin Data Modeler r8\ODBC Reports\Unsupported Sample Reports



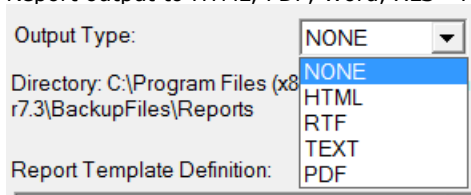
Report Template Builder Reports - Figure 5



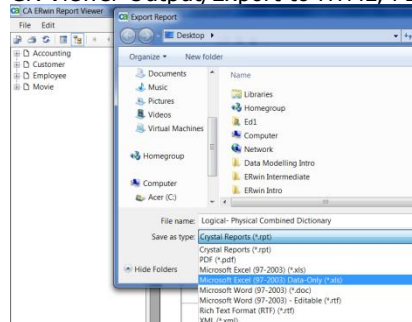
Crystal Developer or Pinned Report - Figure 6



Report output to HTML, PDF, Word, XLS - Figure 8



CR Viewer Output/Export to HTML, PDF Figure 9



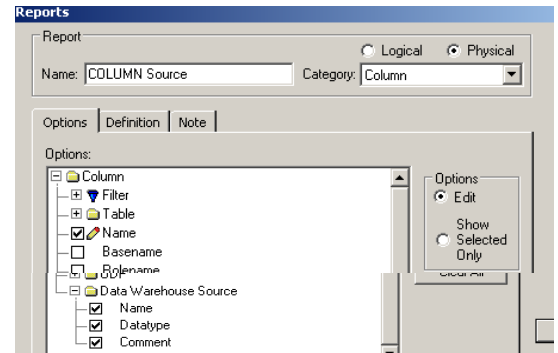
The Data Warehouse Corner...

This issue – ERwin R8 – Data Model Source Meta Data Exposed

Welcome to the Data Warehouse Corner. Each issue we'll identify a business problem facing the Data Warehouse / BI data modeler and discuss solutions and implications for the ERwin modeler. In this issue we'll discuss Data Movement meta data in ERwin R8 and how to report on it using the Erwin SQL Query tool.

In Erwin R7 you had the luxury of using the Data Browser to report Data Source meta data. One would simply open any ERP file (you could even use STANDARD or REPORTS.ERP), define a NEW COLUMN level report, and complete the required fields (see below).

In the R7 ERwin Data Browser you create a New Column level report and select the Data Warehouse Source options at the bottom of the pop-up to report on source column meta data. Available up to version 7.3.X of ERwin

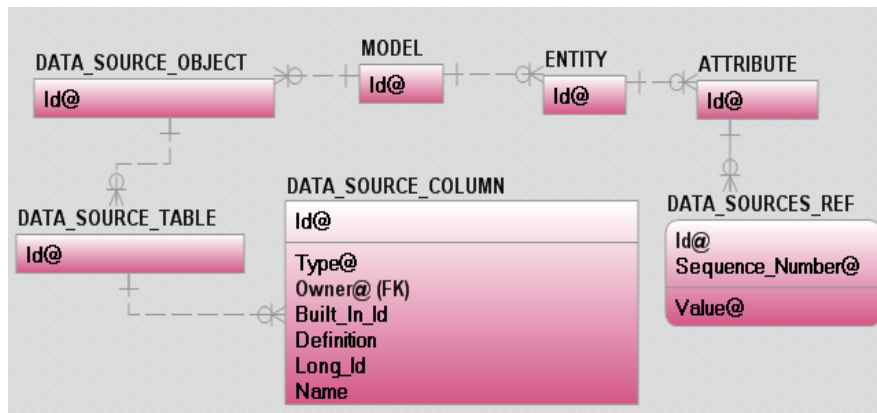


Results of executing the R7 Data Browser report to display Column Source meta data, including the Target Table and Column name, and the Source Schema, Table Name, Column Name, and Source Column datatype.

Table Name	Column Name	Column Data Warehouse Source Name	Column Data Warehouse Source Datatype
EMPLOYEE	EMPLOYEE_KEY	ddw source :CDEMPLOYEE.EMPNO	CHAR(6)
EMPLOYEE	EMPLOYEE_EMAIL_ADDRESS		
EMPLOYEE	EMPLOYEE_PHONE	ddw source :CDEMPLOYEE.PHONENO	CHAR(4)
		ddw source :EMPLOYEE.PHONENO	CHAR(4)
		ddw source :PROJECT.MAJPROJ	CHAR(6)

Note: In 7.3.X Erwin, if you define a TABLE level Data Browser report you'll be able to see DATA WAREHOUSE RULE meta data which defines the population strategy and timing for the TABLE as a whole. We'll pick this up in a future issue...

In Erwin R8, you can achieve the same results, and with a little investigation of the ERwin meta model, you can extend the reporting capability beyond what was offered in R7. Let's start with the meta model...



In the Sandhill 1-day ODBC class we cover in detail the R8 Erwin meta model abstraction and explain Erwin's "dual objects". Two of those dual objects you'll need to know for Data Source reporting are ENTITY and ATTRIBUTE. You'll also need the meta model tables that begin with "DATA_SOURCE_". You might build a Subject Area and Diagram including just these tables so you can focus on the meta data to be extracted. Notice that the MODEL table is the parent of ENTITY, which is the parent of ATTRIBUTE (Entity owns ATTRIBUTE), and there is an identifying relation to DATA_SOURCES_REF. Notice also that MODEL is a parent of DATA_SOURCE_OBJECT which is parent of DATA_SOURCE_TABLE which is a parent of DATA_SOURCE_COLUMN, a very important table here in this discussion.

Start by reviewing the relations to the tables and note the table DATA_SOURCES_REF. This table, like all other tables named "REF" in the ERwin meta model, contains links or pointers to the OBJECT ID's (ID@ and OWNER@ fields) in the parent and child tables that you'll need. In this case DATA_SOURCES_REF links the ID of the COLUMN (ID@) to the DATA SOURCE COLUMN (VALUE@). So from this table we can gather target column and DATA SOURCE column mapping information and any other meta data we want to report.

```
select * from data_sources_ref
```

Results (Row Count = 4):

ID@	SEQUENCE_NUMBER@	VALUE@
859	0	287
859	1	358

At right we have a query against the DATA_SOURCE_COLUMN table that reveals the source column meta data that you select in the Data Source Editor when building the mappings. Here we have selected the Table Name (owner_path), Physical (Column) Data Type, and (Column) Name.

```
select DSC.owner_path, DSC.physical_data_type, DSC.name from data_source_column dsc
```

Results (Row Count = 1):

owner_path	physical_data_type	name
Emoviesnew.ddw source .CDEMPLOYEE	CHAR(6)	EMPNO

```
select ATT.id@, ATT.physical_name, ATT.physical_data_type, ATT.data_source_comment,
       DSR.value@
from attribute ATT, data_sources_ref DSR where ATT.id@ = DSR.ID@
```

Results (Row Count = 2):

id@	physical_name	physical_data_type	data_source_comment	value@
839	EMPLOYEE_KEY	CHAR(18)	substr(empno, 1,5)	292
859	EMPLOYEE_PHONE_NO	INTEGER	Select the Phone information from EMPLOYEE ...	287

Next we join the ATTRIBUTE table to the DATA_SOURCE_REF table to reveal the data source transform comment and the column meta data including target Column ID (id@), Target Physical (Column) Name, (target column) Physical_Data_Type, comment, and Source Column ID (value@)

Finally we pull in the Target Table Name (from the ENTITY table – remember dual objects), and then pretty up the report-column headings with some correlations. We now have a complete report of the Source Table/Column and Target Table/Column mapping and transform comments/instructions.

```
select TBL.PHYSICAL_NAME AS "TGT TBL NAME",
       ATT.physical_name as "TGT COL NAME",
       ATT.physical_data_type as "TGT COL DATA TYPE",
       DSC.owner_path AS "SRC TABLE",
       DSC.name AS "SRC COL NAME",
       DSC.physical_data_type AS "SRC COL DATA TYPE",
       ATT.DATA_SOURCE_COMMENT AS "XFORM COMMENT",
       ATT.id@ as "TGT COL ID",
       DSR.VALUE@ as "SRC COL ID"
from attribute ATT inner join data_sources_ref DSR on ATT.id@ = DSR.ID@
left outer join data_source_column dsc on dsc.id@ = value@
left outer join entity TBL on tbl.id@ = att.owner@
```

Results (Row Count = 2):

TGT TBL NAME	TGT COL NAME	TGT COL DATA TYPE	SRC TABLE	SRC COL NAME	SRC COL DATA TYPE	XFORM COMMENT	TGT COL ID	SRC COL ID
EMPLOYEE	EMPLOYEE_KEY	CHAR(18)	Emoviesnew.ddw source .CDEMPLOYEE	EMPNO	CHAR(6)	substr(empno, 1,5)	839	292
EMPLOYEE	EMPLOYEE_PHONE	INTEGER	Emoviesnew.ddw source .CDEMPLOYEE	PHONENO	CHAR(4)	Select the Phone informatio...	859	287

Here is the complete query that you can run against your own models and export to Excel or use Crystal Reports to create a more elaborate presentation. You'll have to do a little extra work where multiple source columns are mapped, and there are a few ways to accomplish this...

```
select TBL.PHYSICAL_NAME AS "TGT TBL NAME",
       ATT.physical_name as "TGT COL NAME",
       ATT.physical_data_type as "TGT COL DATA TYPE",
       DSC.owner_path AS "SRC TABLE",
       DSC.name AS "SRC COL NAME",
       DSC.physical_data_type AS "SRC COL DATA TYPE",
       ATT.DATA_SOURCE_COMMENT AS "XFORM COMMENT",
       ATT.id@ as "TGT COL ID",
       DSR.VALUE@ as "SRC COL ID"
from attribute ATT
       inner join data_sources_ref DSR on ATT.id@ = DSR.ID@
       left outer join data_source_column dsc on dsc.id@ = value@
       left outer join entity TBL on tbl.id@ = att.owner@
```

Now that you have defined the Data Movement mapping for each of the Tables and Columns in the model, you can turn your attention to publishing this meta data using the SQL Query tool and a little knowledge of the ERwin meta model. You can use the grouping and subselect ability of Crystal to filter data from the final report, and you can publish the report to HTML using Crystal as well.

Next issue: Model Meta Data integration into third party tools....