ERwin Times

SANDHILL

Q2 Your #1 ERwin Newsletter Focused on ERwin

Hello Readers

I am not sure if many of our readers have had the chance to visit Las Vegas, the location of CA World 2010.

For those that have, I am sure they will agree that CA World 2010 was a great event. One where clients got to meet with CA Executives, ERwin development staff, Sandhill team members and of course each other.

The most exciting story to be told at this years CA world is ERwin's upcoming R8 and the amount of work that is going to make this everything clients are looking for and more. Check out the CA world article for a detailed review and links to recent webcasts.

The Sandhill team has been on the road again, with trips to New York, Ottawa, Hartford, Boston and Detroit, just to mention the most recent trips. We still have more events and trips in the works. Drop us a line if you are interested in having us drop by and we will let you know when we are planning to visit your area.

As always part of this column is to inform you of the latest version of ERwin. As of this time it is V7.3.9 (build 2386).

If you would like a copy of this build and you are on maintenance, please let us know.

Regards

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

CHANNEL PARTNER
ENTERPRISE SOLUTION PROVIDER

CA World 2010 - Update

Putting on an event such as CA world is a huge task, getting speakers, arranging sessions, working with clients and partners. It is no easy feat, yet CA masters this very successfully every time. The valuable ERwin sessions are put together through the tireless efforts of unsung heroes and are eagerly awaited by users local and afar. So, with CA World 2010 ended, we can tell you that it was another very successful event. There were good sessions and great interactions with CA, Clients and Partners.

Of course, everyone was looking forward to the ERwin State of the Union Address. If you have not heard already, then over the next few months you will be hearing a lot about CA's upcoming ERwin R8. The attendees had the chance to hear Sr Product Manager Danny Sandwell talk about R8. He also talked about what CA are planning strategically beyond R8 and their commitment to expand support to help build a stronger ERwin community.

While most of the presentations (ppt's) have been loaded up to the CA World website (A link can be found in our last newsletter) The state of the Union ppt is not yet available. If you would like to obtain a copy, please email Sandhill and we will work with you and CA to get you a copy.

Alternatively, during a recent ERwin User Group meeting in NY, CA's Danny Sandwell delivered a PLC Community webcast on the ERwin State of the Union Address. This webcast was recorded.

Click Here To Access The Replay:

https://www.livemeeting.com/cc/cai/view?id=75298W&role=attend&pw=n*-pd5DKP

As everyone knows, Sandhill is exclusively focused on ERwin and we constantly strive to promote ways of interacting with the ERwin community. In our past newsletters we have talked about www.ERwin.com and all the interaction that you, as a data modeler, can have with CA and its partners. If you are not part of the ERwin community yet there are plenty of ways to join in.

Sandhill supports and contributes to CA's expansion of the community concept. Today, every data modeler can be part of a community not only locally but nationally and internationally. Check out the following new link. Once you register you will be able to join the communities of ERwin around the world.

http://www.ca.com/us/communities.aspx

Speaking of heroes, here is your chance to become a data modeling hero. If you want your own "official ERwin T-Shirt" and the chance to win an Ipod Touch, then all you have to do is submit your ERwin tip or technique that helps make your job as a data modeler easier. Click on the link below to submit your tip.
http://erwin.com/tips_techniques/contest/

For those interested in reviewing the latest tips and techniques that have posted, click on the following link.

http://erwin.com/tips_techniques/

As we say good bye to Las Vegas and the "ding, ding" of the slot machines fade way, we are reminded that next years CA World is only one year and 4 months away. CA have announced that it will be hosting its next User Group conference on October 16 – 20, 2011

See you there? ☺ Robert

Customizing the Forward Engineering Template (FTE)

The guys at Sandhill get asked many "How do I?" questions but some make us think more than others. Often questions relate to workflow and the automation of repeated steps. Those about physical modelling can sometimes be accommodated by customising the Forward Engineering Template (FET).

FET's were introduced to ERwin in V7.0 but their usability greatly improved in V7.3 when the new TLX macro language and its Integrated Development Environment (IDE) arrived. FET's are dbms specific templates which control the generation of DDL and are written in the TLX macro language. They can be edited using the template Editor accessible from the Forward Engineering Editor (choose the Edit button).

The TLX macro language is more procedural than the old EML language (still used for naming, scripts, triggers, etc) and will replace it in time. The TLX language is also more generic, has a reduced instruction set, contains layout macros and allows the user to control the context. The context identifies which particular object is in scope. This was not possible with the EML language where the macros were very specific. The EML macros require a specific type of object to be in context and will then perform a prescribed action. It is not possible in the EML to programmatically move the context from the current object to a referenced object. As an example consider a situation where the context is a FK Key Group/Index and the user wished to move to the associated relationship, read its FK Constraint name before returning to the original object. In the old EML language this sort of control is not possible whilst it could be achieved quite simply in the TLX language using the following code.

```
/* Comment – The initial context object is a FK Key_Group object.*/

PushReference("Relationship_Ref")
Property("Physical_Name")
Pop
```

The line PushReference("Relationship_Ref") uses the PushReference() macro and the key_Group property "Relationship_Ref" which holds the internal identifier of the associated relationship. This property works in a similar manner to a FK and the combination moves the referenced object into context. What actually happens is the referenced relationship object is pushed onto the top of the stack of objects ERwin is working with and above the Key_Group object which was initially in context. Whichever object is at the top of the stack is in context (has focus).

The line Property("Physical_Name") reads and outputs as a string the value of the property named in the parenthesis. This is the generic macro for accessing property values. In this case the relationship's Physical_Name property corresponds to the physical FK constrain name.

The last line removes which ever object is currently at the top of the stack moving the object below it into context. In this example context returns to the original Key_Group object.

One example of this type of request was from a customer using SQL Server 2005 where about 80% of their indexes are placed in a single filegroup (SQL Server storage object) called FG_Index. This required the same repetitive assignment task on a large number of indexes. The solution was to automate index assignment to the default during DDL generation where no specific filegroup had been assigned in the model leaving the users to focus on the exceptions. This was achieved by adding a model level UDP and customising the FET for SQL Server 2005.

A model level UDP called Index_FGroup was added to the physical side of the model to hold the name of the default filegroup and was given the default value FG_Index. If the UDP is defined as a list type with a single value this prevents the value being easily changed. When retrieving V7 UDP values using the TLX macro language (and the ODBC interface or the API) they appear as properties with a three part naming convention of *Object_Type.Logical/Physical.UDP_Name*. In the above case the property holding the default filegroup name will be at the model level and will be named Model.Physical.Index_FGroup. Provided the correct object is in context retrieving its value using the TLX language requires the generic Property() macro as shown below.

```
Property("Model.Physical.Index_FGroup")
```

As with editing any code there is some investigation required to work out what changes need to be made and where to make them.

The FET files contain a number of templates each of which perform a specific function. For example the template "Create Entity" manages the generation of CREATE TABLE statements but may call other templates such as "Clause: PKConstraint" or "Create Key_Group". Using the Find tool to search a template for macros such as "FE::Option()" (Checks if a specific FE option has been selected) and "Execute()" (Calls another template) can assist in working out the flow and where edits are required. In this case the "Create Entity" template calls the "Create Key_Group" template which contains the follow snippet of code.

```
[                                       /* Comment – Start of conditional block*/
        "\n"                            /* Comment – new line*/
        "ON "                           /* Comment – Quoted therefore a  literal */
        <                               /* Comment – Start of Propagating Block */
                [                       /* Comment – Start of conditional block*/
                 PushReference( "SQLServer_File_Group_Ref" ) /* Comment – Moves context to referenced fielgroup*/
                 "\""                   /* Comment – Use escape character to generate an opening quote*/
                < Property( "Name" ) > /*Comment – Emit name of current object*/
                "\""                    /* Comment – Use escape character to generate a closing quote*/
                Pop                     /*Comment – Remove the current context object from the stack*/
                ]                       /* Comment – Close conditional block
        >                               /* Comment – Close Propagating block
]                                       /* Comment – Close conditional block
```

There is neither sufficient space here to cover all aspects of the TLX syntax nor the workings of FET templates but this code is only executed if the “Physical Storage” option has been selected in the FE dialog and will only generate SQL if an index in the model has a specific filegroup assignment. The full syntax of the TLX language is covered in documentation automatically installed with ERwin since the release of V7.3.0.

The above snippet needs to be changed so that any Index filegroup assignments in the model are used as at present but to use the default filegroup where no specific assignment is made. One method of achieving that is to turn the above snippet into an “If, Then, Else” statement. The first part (the If statement) shown below comes into play where no specific filegroup assignment is made picking up the name of the default filegroup instead.

```
[                                               /* Comment – Start of conditional block*/
        "\n"                                    /* Comment – new line*/
        "ON "                                   /* Comment – Quoted therefore a  literal */
        @if (IsPropertyNull("SQLServer_File_Group_Ref") ) /* True if no filegroup assignment made in model for index*/
        {
                PushOwner                       /* Moves context to object Owner, IE its entity/table*/
                PushOwner                       /* Moves context to tables owner, IE the root model object*/
                "\""                            /* Generates opening quote
                < Property( "Model.Physical.Index_FGroup" ) >  /* Gets the value of the UDP*/
                "\""                            /* Generates a closing quote
                Pop                             /* Removes model from stack*/
                Pop                             /* Removes entity/table from stack.
        }
```

The original code now forms the second part (the Else statement) which remains unchanged. This handles the physical storage clause for those indexes which have a specific filegroup assignment.

```
        @else
        {
                PushReference( "SQLServer_File_Group_Ref" )
                "\""
                < Property( "Name" ) >
                "\""
                Pop
        }
]
```

The example below shows one table (ORDER_ITEM) where no filegroup assignment has been made so the default filegroup has been used and another (SALES_ORDER) where a specific filegroup assignment to FG_001 has been made.

```
CREATE UNIQUE INDEX XPKORDER_ITEM ON ORDER_ITEM
(       item_no         ASC,
        order_no        ASC)
ON "FG_INDEX"
go

CREATE TABLE SALES_ORDER
(       order_no        char(18)  NOT NULL,
        order_date      char(18)  NULL,
        order_status    char(18)  NULL,
        order_value     char(18)  NULL,
        customer_no     char(18)  NULL)
go

CREATE UNIQUE INDEX XPKSALES_ORDER ON SALES_ORDER
(
        order_no        ASC
)
ON "FG_001"
go
```

DefaultIndexFileGroup

Specific FileGroup Assignment for this index

Unlike the EML language the TLX language can access all model data and achieves its greater power with a reduced instruction set. However, to exploit that power users require an understanding of the models data structure. Many ERwin users have no desire to develop that knowledge in house and use Sandhill to manage their FET customisations together with their ODBC and API requirements.

Note. The template code in some builds may differ from the example provided as software fixes are sometimes effected by changes to the FET

The Data Warehouse Corner…

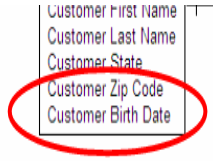
This issue - Slowly Changing Dimensions…who said they were slow?

Welcome to the Data Warehouse Corner. Each issue we'll identify a business problem facing the Data Warehouse / BI data modeller and discuss solutions and implications for the ERwin modeller. In this article we'll discuss Slowly Changing Dimensions and their impact on BI applications.

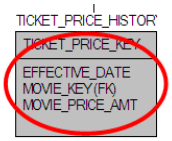
This author first saw the impact of slowly changing dimensions when the team was building data marts to track insurance sales for each month. The sales organization structure would be re-built from the source system (we had a single tier DW system which meant that the production system data was directly input to the data mart) to align with the sales for that month. Within a given month we had integrity within the snapshot data, but if we looked back into the historical sales data using the existing organization structure, we were faced with problems. When we attempted to analyse current sales data using the historical organization hierarchy, we ended up with orphan salespeople (no reporting office), or sales offices that had a revenue position when they weren't even in existence! As if we didn't have enough data quality problems, we created some of our own!!!

When we first examined the problem we discovered that Ralph Kimball's classic interpretation of this problem, that is being caused by a Slowly Changing Dimension, was correct. In our case it wasn't 'slow' (referring to the inconsistent timing of the change), but rather episodic, in that they would change many times within a short time frame, or not change at all for many months. The ETL team was victimized each month as they didn't know when the change was to happen, to better deal with it.
To address the problem we identified the changing nature of the data in specific ways, in order to better deal with changes over time. The characterization was done as follows (a sample instance in parenthesis) and can be characterized as follows:

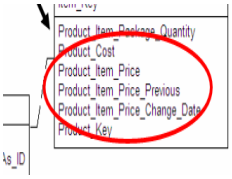
Type 1 (birth date) – no temporal association. This easiest of structures is where the current data represents/is the value that has been associated with the tuple since its inception. (Little causal analysis can be constructed, rather only an evaluation of the current or most recent business cycle is accurate.)



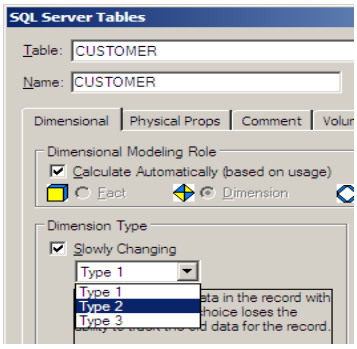
Type 2 (product price) – the value at each instance and a temporal association representing the date that the value has changed (for analysis purposes – not necessarily the actual date of the change. This is the most popular of characterizations but can be a performance challenge.



Type 3 (smoker indicator) - only the current value and previous value is of interest. Any additional prior values are of little interest for analysis purposes. Can have some causal analysis value when trying to understand the reasons behind a change in value/state of a measure.



To help the analyst, Erwin has the capability to display this type of characterization on the Physical Side of the model when the Dimensional notation is chosen (Model…Model Properties..General and Notation tabs).
This is helpful to the data analyst and application builder since Data Browser reports can be produced to support the characterization and data mapping to be performed behind the process of loading the data.
This is also helpful to the DBA since it can also indicate where additional database objects can lead to improved performance, such as VIEWS, MQT's or Indexes.



In this article we've extended the data model by illustrating how data changes over time. This is valuable to the data analyst as it defines analysis potential, and for the data warehouse DBA as it gives a better understanding of data growth and where performance issues may arise.

Next issue: Data Model and Source Data Quality - definition and assessment…

Article by Joe Cullen, Sandhill Consultants
Joe.Cullen@sandhillconsultants.com