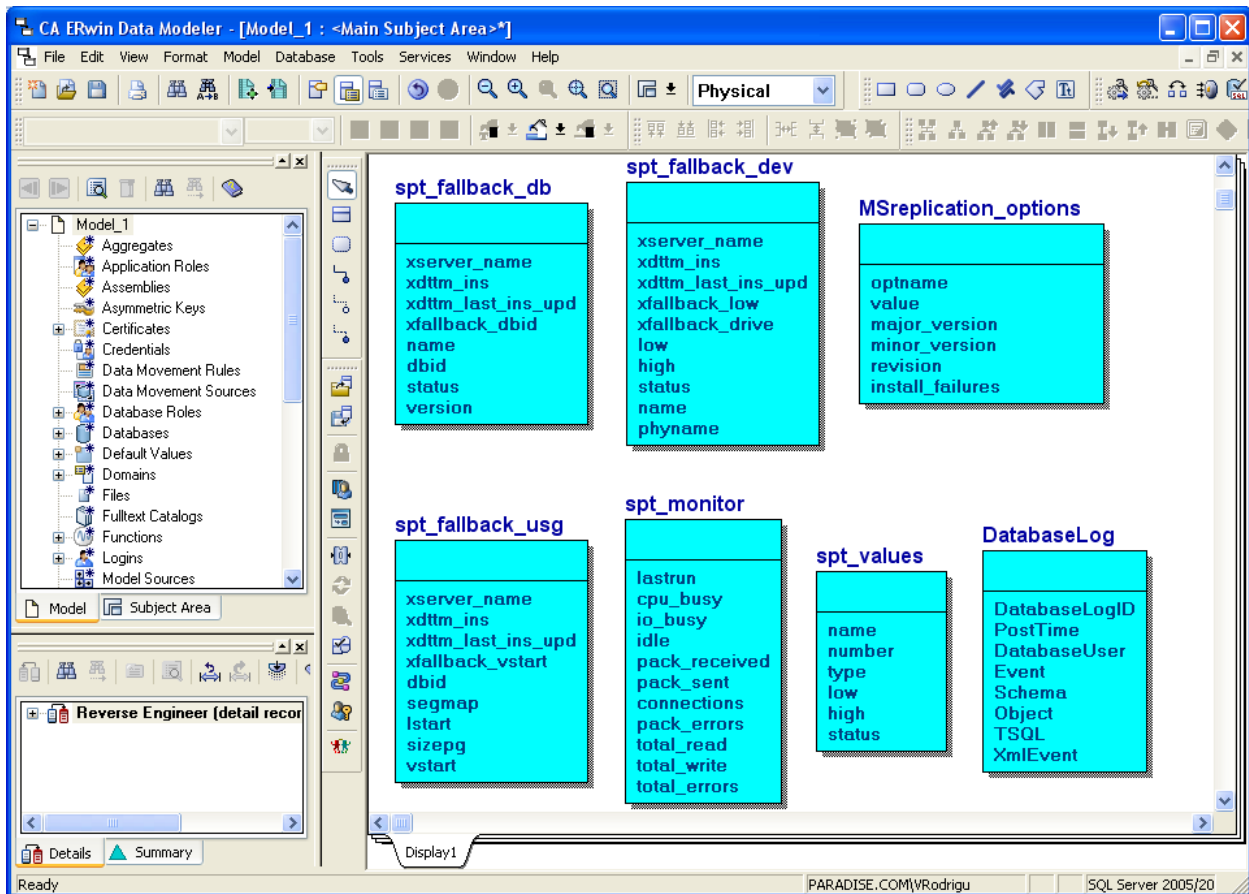




Customizing the ERwin FET file to Automatically Remove SQL Server Collate Syntax

By Victor Rodrigues

Often, after reverse engineering a SQL Server 2005 or 2008 database environment, users find that the tool has collected the COLLATE property on many of the tables. Case in point, I have Reverse engineered the default 'master' database on a SQL Server 2005 server:



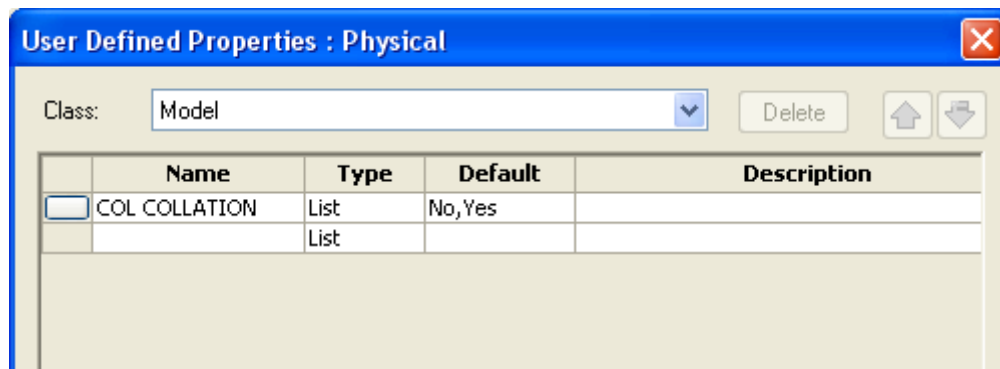
If I preview the SQL generated for the table `spt_fallback_dev` it looks like:

```
CREATE TABLE spt_fallback_dev
(
    xserver_name          varchar(30) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,      xdttm_ins
datetime NOT NULL ,
    xdttm_last_ins_upd   datetime NOT NULL ,
    xfallback_low        int NULL ,
    xfallback_drive      char(2) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    low                  int NOT NULL ,
    high                 int NOT NULL ,
    status               smallint NOT NULL ,
    name                 varchar(30) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    phyname              varchar(127) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL
)
go
```

The COLLATE syntax is valid and does exist on these objects. However, usually the COLLATE is defined as a server default and it is unnecessary and even improper to include it in the table definition since it may conflict with the environment to which we will be generating the script.

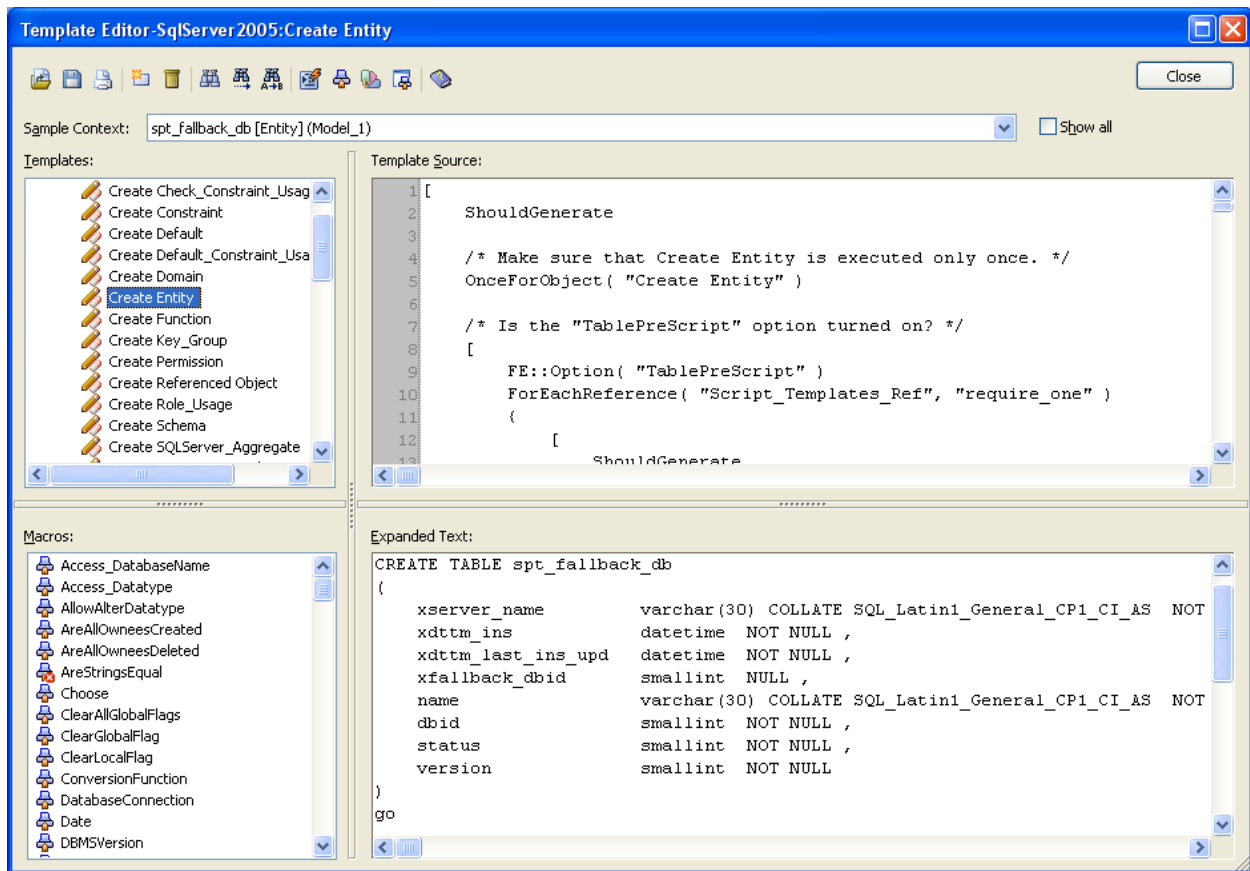
We will now create a simple solution for this using a Model Level UDP and a handful of added lines of code in the FET editor.

First, we define the UDP at the model level. It will be a List type UDP with values of No and Yes.



Next we will delve into the Forward Engineer Template editor. If you are not familiar with this editor, there is a demo video in the Videos folder of the ERwin Install path. This video is also available On Demand on the www.ERwin.com website.

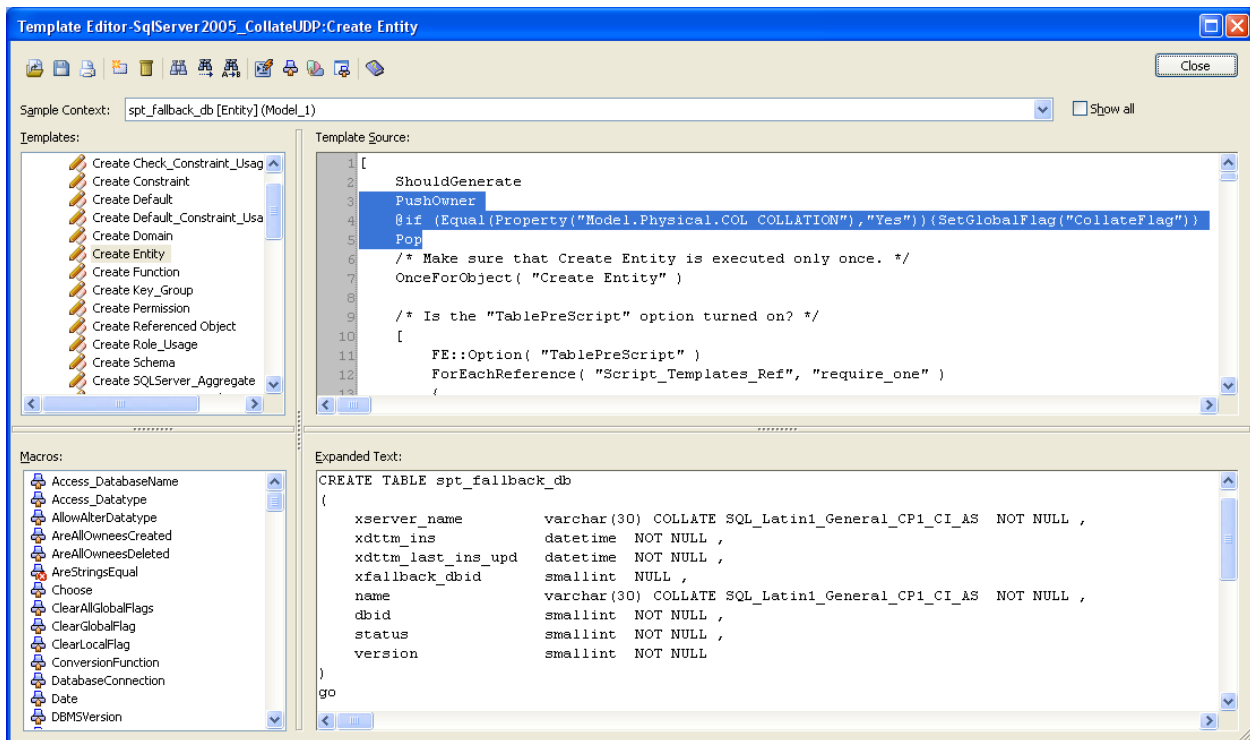
In the FET editor select your sample Context to be the same table, `spt_fallback_dev`. Select the Create Entity from the list of templates. Notice that the Expanded Text includes the Collate statements.



Next, we are going to define a Global Flag depending on whether the UDP is set to yes or no. We do this by entering the following text into the Template Source editor right between ShouldGenerate and the beginning of the first code comment.

```
PushOwner
@if (Equal(Property("Model.Physical.COL_COLLATION"), "Yes")) {SetGlobalFlag("CollateFlag")}
Pop
```

Now our FET code looks like...

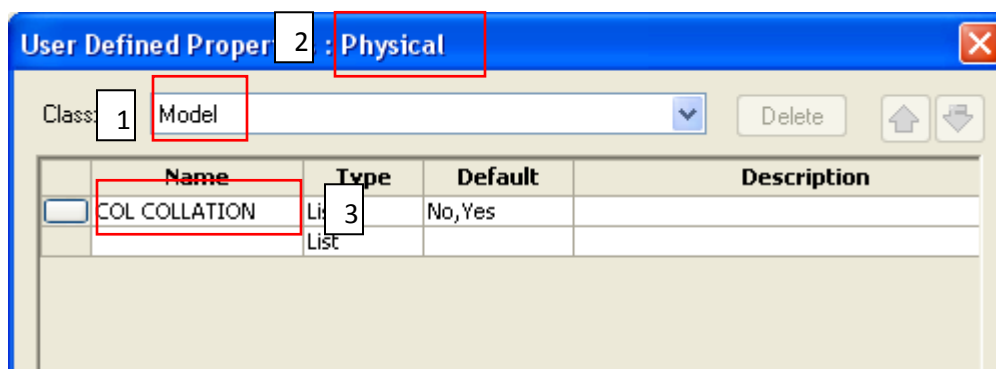


So what does that all mean?

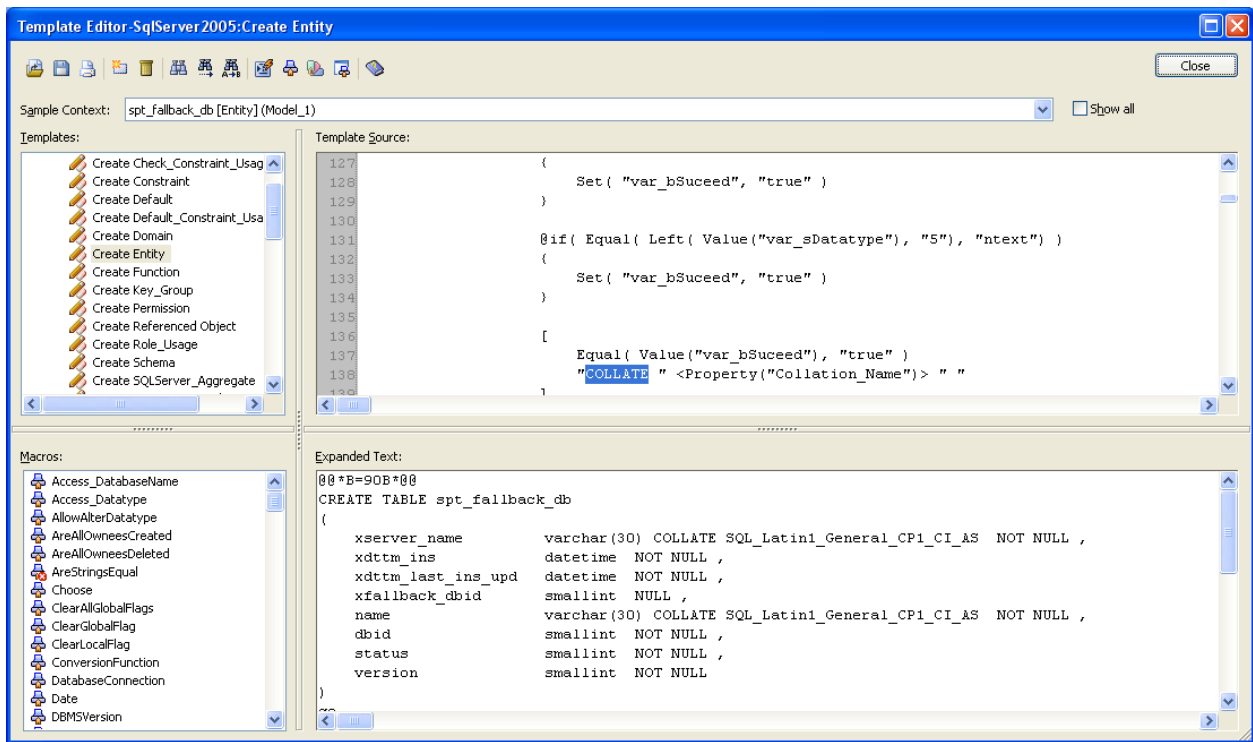
First of all, we want to get to the model level but we are in the Entity so we PushOwner. The FET works like any stack and we push each time we want to climb a level. If you have ever used the ErwinSpy (and it is a must if you plan to do anything with the FET or the API) you can see exactly how many layers in the stack one object is from another.

Next, we have an if condition checking for equality between the UDP and the string “Yes”. If they match we set the global flag variable “CollateFlag” (the name is arbitrary).

The trickier part of the syntax is the UDP name. In general all model properties are simply referenced like Property(<name>”). But in the case of the UDPs the syntax is (1)<object_level>.(2)<logical or physical model side>.(3)<UDP Name>. So we end up with Model.Physical.COL COLLATION



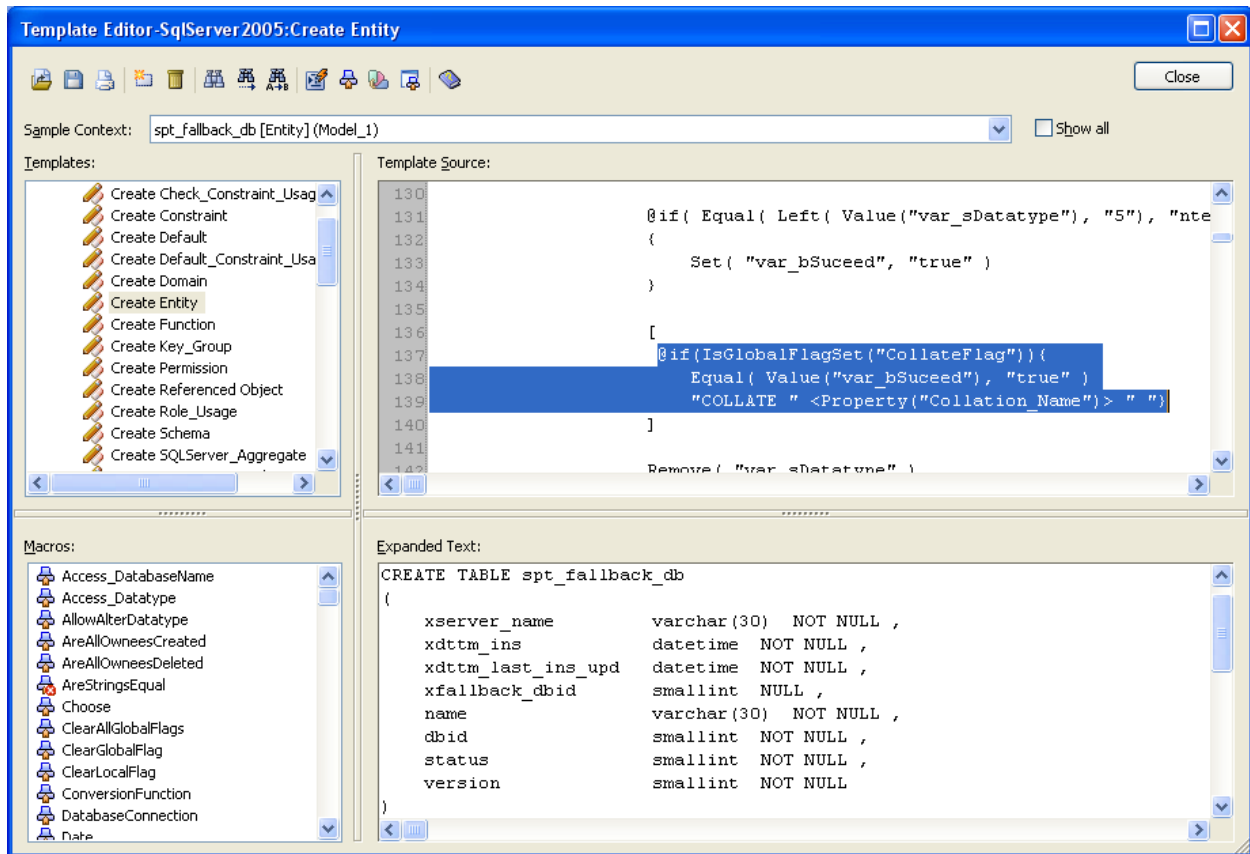
We finish off this clause with a Pop statement to move back from our push position down to the Entity level. That is the scariest bit of it. We have just one step to go. Use the Find in the Template editor to locate the key word COLLATE. If you use a case sensitive match you will see this:



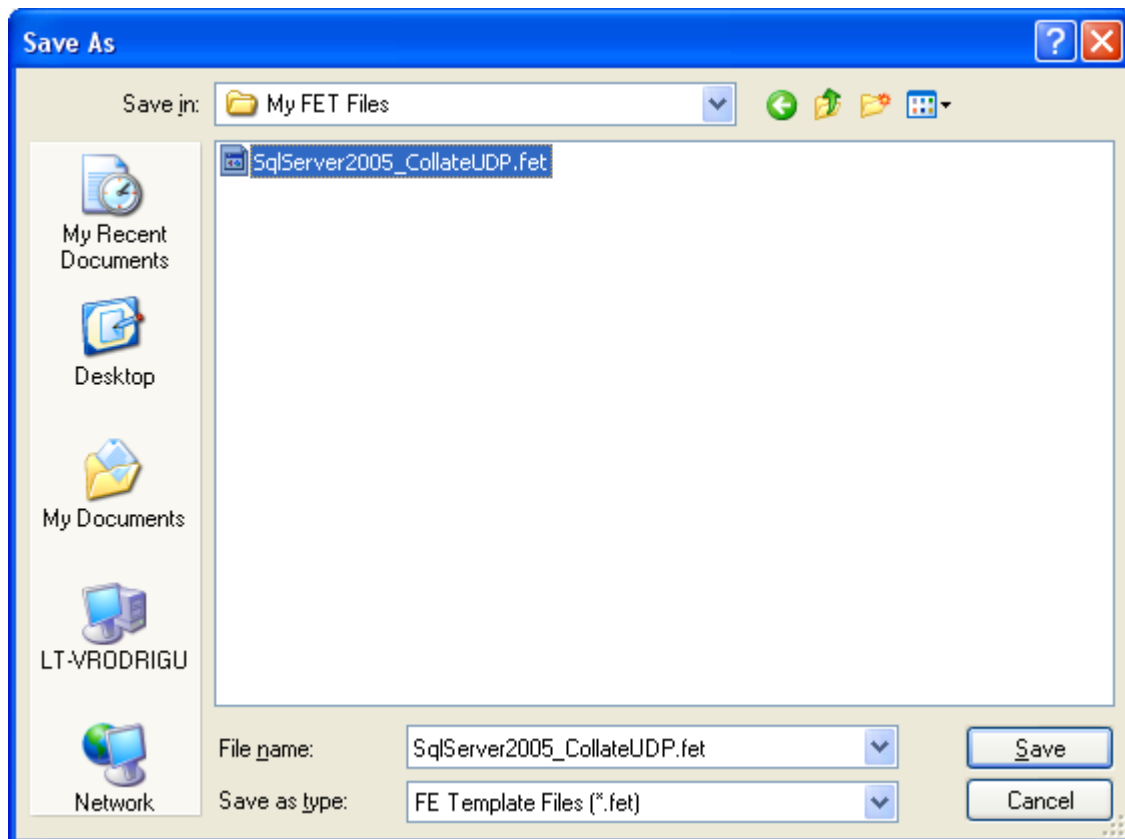
This is the code that is generating the COLLATE syntax. We need to make it conditional. We are going to do this by placing the entire thing within the if condition...

```
@if(IsGlobalFlagSet("CollateFlag")) { ... }
```

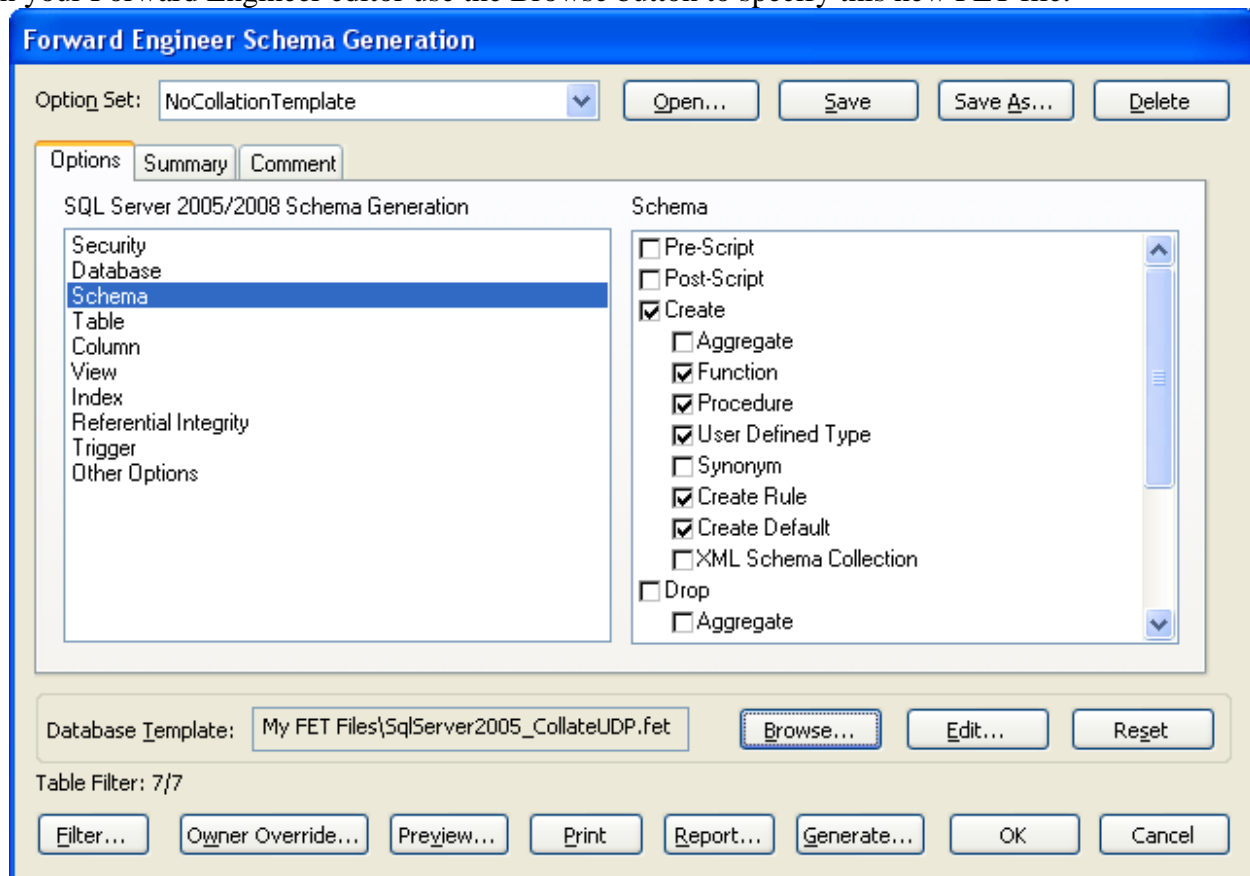
Now our editor looks like:



Make sure to enclose the entire collate syntax within the {}. The IsGlobalFlagSet is a default macro that does exactly what you expect, it returns TRUE if the specific value is defined. Notice that the table definition has already lost its COLLATION syntax (provided you left the UDP set to No). Save the template file to your local drive. You cannot override the Default but you can save this as your standard FE template file once you have saved it to your drive.



Now, in your Forward Engineer editor use the Browse button to specify this new FET file:



Preview your good work:

