

Using Oracle Stored Procedures in a .NET Entity Model

- Basic Outline -

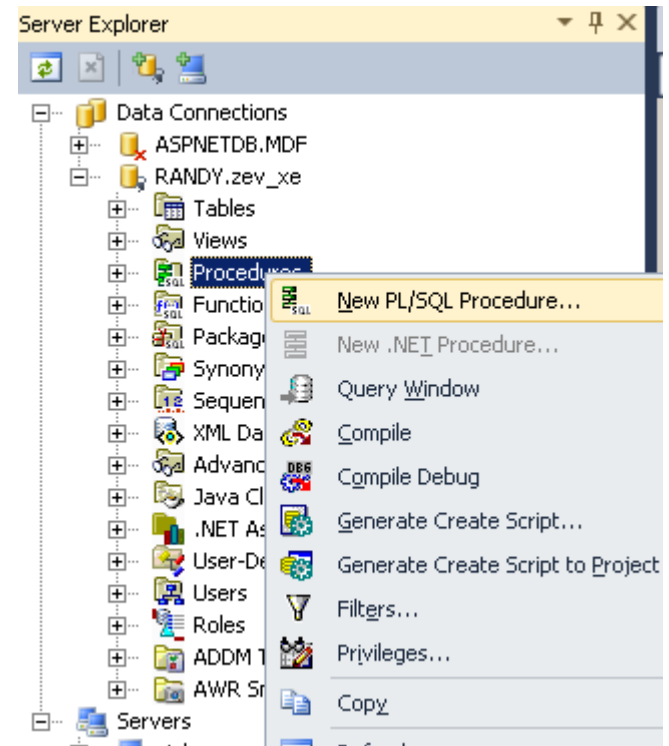
Create a PL/SQL Procedure

Import into Entity Model

Call Procedure

Create a PL/SQL Procedure

- Visual Studio can create and edit PL/SQL Procedures
- Native Oracle Tools also can be used if preferred



Example PL/SQL Procedure

- This procedure lookups up an artist record by name.
- If the artist cannot be found, it creates a new record.
- The ID of the artist is returned to the caller.

```
PROCEDURE "GET_ARTIST" (  
  "ARTISTNAME" IN VARCHAR2,  
  "ARTISTID" OUT INT) IS  
  
BEGIN  
  
  select ARTIST_ID into ARTISTID  
  from ARTIST  
  where ARTIST_NAME = ARTISTNAME;  
  
EXCEPTION  
  when no_data_found then  
    insert into ARTIST(ARTIST_NAME) values(ARTISTNAME)  
    returning ARTIST_ID into ARTISTID;  
  
END;
```

The .NET Entity Model supports PL/SQL procedures, but not functions.

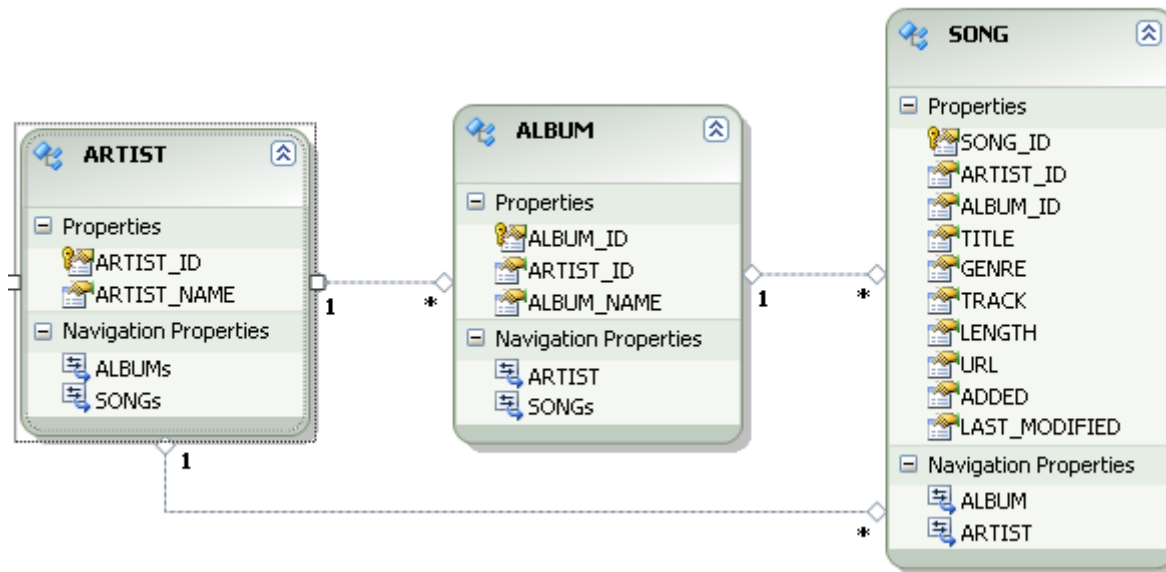
Another Example Procedure

- This procedure lookups up an album record by its artist id and album name.
- If the album cannot be found, it creates a new record.
- The ID of the album is returned to the caller.

```
PROCEDURE "GET_ALBUM" (  
  "ARTISTID" IN NUMBER,  
  "ALBUMNAME" IN VARCHAR2,  
  "ALBUMID" OUT INT) IS  
  
  BEGIN  
  
    select ALBUM_ID into ALBUMID  
    from ALBUM  
    where ARTIST_ID = ARTISTID and ALBUM_NAME = ALBUMNAME;  
  
  EXCEPTION  
  
    when no_data_found then  
      insert into ALBUM(ARTIST_ID,ALBUM_NAME) values(ARTISTID,ALBUMNAME)  
      returning ALBUM_ID into ALBUMID;  
  
  END;
```

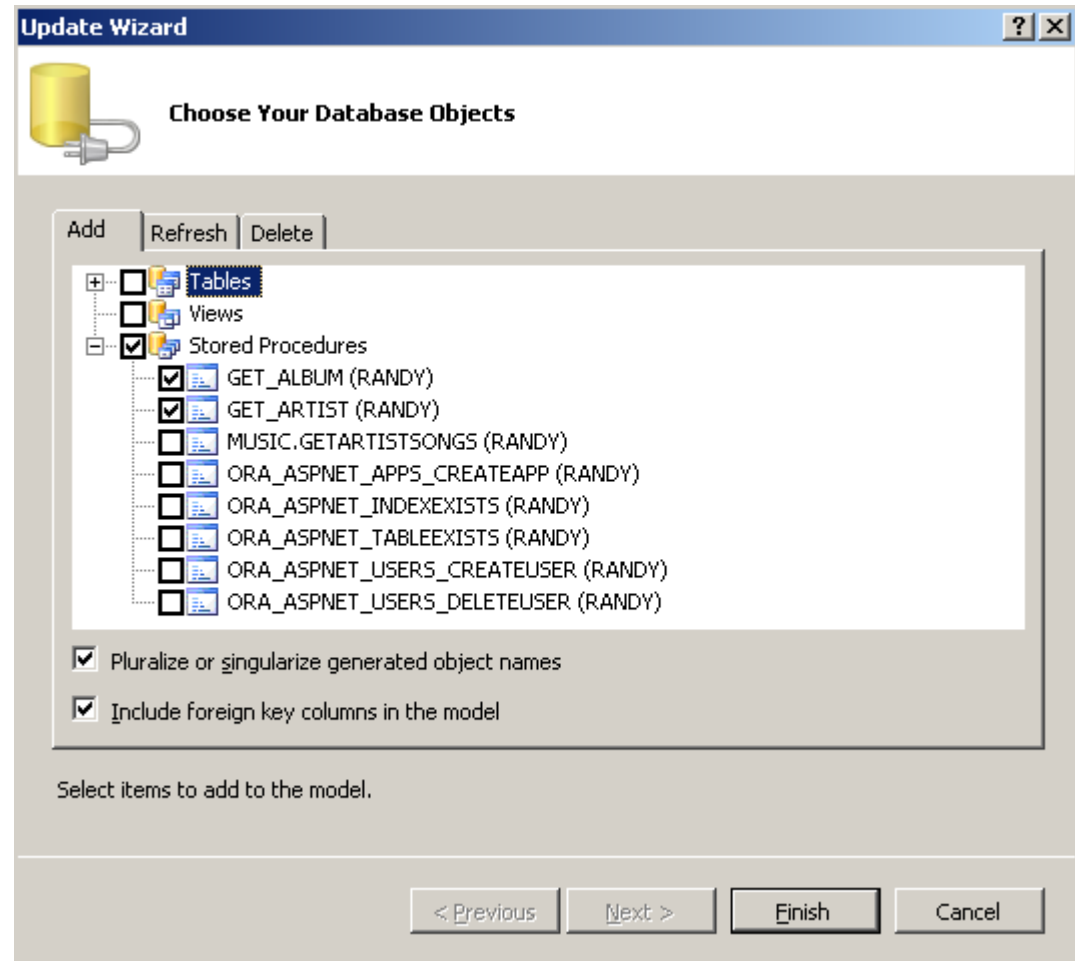
The Entity Model

- Create a new Entity Model, then import any required tables using the Wizard.
- This part of the model makes data binding easier.



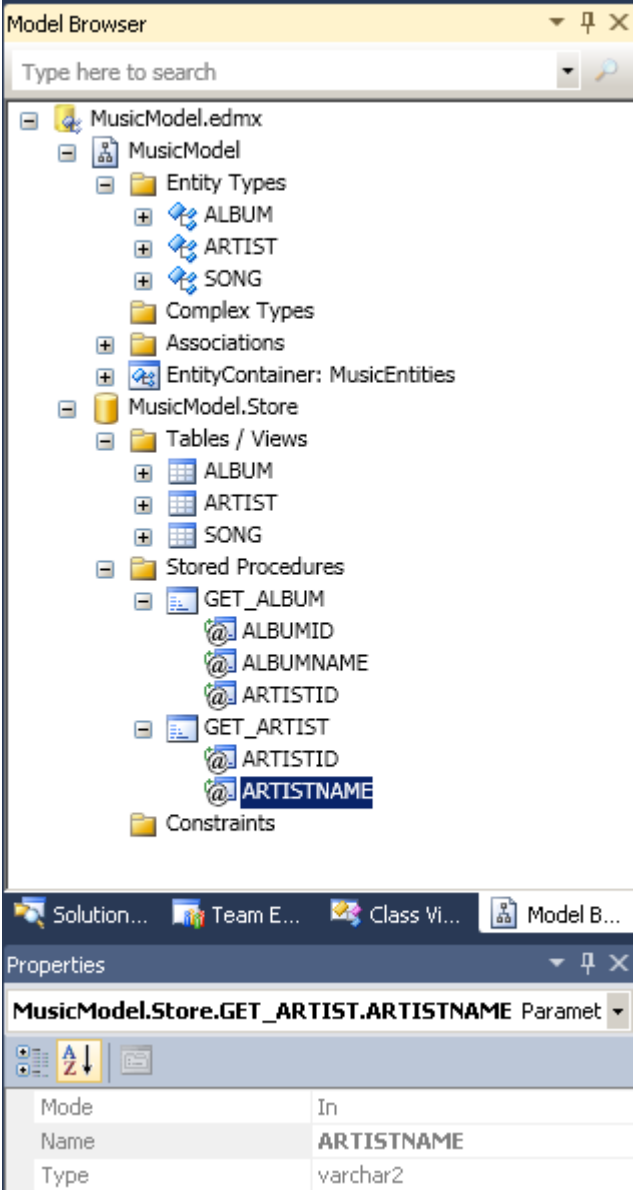
Importing Procedures

- The Update Wizard will import the procedures.
- You can also import new ones later.
- Just right-click the model and select *Update Model from Database* to run the Wizard again.



Model Procedures

- Imported procedures are now part of the model.
- Parameter names and types are automatically determined.
- The MusicModel.Store isn't intended to be used directly though.



The screenshot displays the Model Browser and Properties windows. The Model Browser shows the structure of the MusicModel.edmx, including Entity Types (ALBUM, ARTIST, SONG), Complex Types, Associations, EntityContainer: MusicEntities, MusicModel.Store (Tables / Views: ALBUM, ARTIST, SONG; Stored Procedures: GET_ALBUM, GET_ARTIST), and Constraints. The Properties window shows the details for the parameter @ARTISTNAME in the GET_ARTIST procedure, including its mode (In), name (ARTISTNAME), and type (varchar2).

Mode	In
Name	ARTISTNAME
Type	varchar2

Mapping Procedures to Functions

Add Function Import [?] [X]

Function Import Name:
GetAlbumID

Stored Procedure Name:
GET_ALBUM

Returns a Collection Of

None

Scalars: []

Complex: [] [Update]

Entities: []

Stored Procedure Column Information

[Get Column Information]

Click on "Get Column Information" above to retrieve the stored procedure's schema. Once the schema is available, click on "Create New Complex Type" below to create a compatible complex type. You can also always update an existing complex type to match the returned schema. The changes will be applied to the model once you click on OK.

[Create New Complex Type]

[OK] [Cancel]

- Right-click the procedure and select *Add Function Import...*
- Some procedures may need these options, but this example does not
- Click OK to create a new function in the model that is mapped to the PL/SQL code.

Calling a Stored Procedure

```
private decimal GetAlbumID(decimal artistID, string albumName)
{
    using (MusicEntities db = new MusicEntities())
    {
        System.Data.Objects.ObjectParameter output =
            new System.Data.Objects.ObjectParameter("ALBUMID", typeof(int));
        db.GetAlbumID(artistID, albumName, output);
        return (decimal)output.Value;
    }
}
```

- Having a method in the model makes it very easy to use PL/SQL procedures.
- While the OUT parameter requires some special handling, that is not difficult to setup either.

Stored Procedures in Entity Models

- Summary

- Run the *Update Model from Database* wizard to import stored procedures
- Use the *Add Function Import* wizard to map procedures to model methods
- Call the methods after obtaining a model context object.