SD10672: Exploring Entity Framework

Ben Rand

Director of IT, Job Industrial Services, Inc.

Twitter: @leadensky

Class summary

This class provides an introduction to Microsoft's Entity Framework, one of the leading object-relational mapper frameworks in use today. With Entity Framework, developers can focus more of their effort on dealing with how objects in their problem domain relate to each other, and less effort worrying about data-persistence code. The class starts by taking a quick look at "Database First" creation of an Entity Framework model. It then moves to the "Code First" style of development, including POCO classes, DbContext configuration. We learn how to consume the resulting Entity Framework model in various different project types (including a Console app and an AutoCAD plug-in application).

Who Am I?

- Director of IT, Job Industrial Services (Salt Lake City, UT)
  - Developer
  - Servers/VMWare/Network
  - Helpdesk
  - CAD Manager
- Languages
  - LISP, VBA, VB6, VB.NET, **C#**, Java, JavaScript, Ruby
- Autocad since R12, programming since ~1996
- Industrial Engineering field since 2001

Be heard! Provide AU session feedback.

- Via the Survey Stations, email or mobile device.
- AU 2016 passes awarded daily!
- Give your feedback after each session.
- Give instructors feedback in real-time.

More Questions? Visit the AU Answer Bar

- Seek answers to all of your technical product questions by visiting the **Answer Bar**.
- Open daily 8am-10am and Noon-6pm and located just outside of Hall C on Level 2.
- Staffed by Autodesk developers, QA, & support engineers ready to help you through your most challenging technical questions.

Forget to take notes? No problem!

After AU visit:
AutodeskUniversity.com

Click on **My AU** to find:

- Class Recordings
- Presentations
- Handouts

All of your sessions will be there to enjoy again and again.

Key Learning Objectives

Create EF model using DatabaseFirst or CodeFirst

Configure Database options using DataAnnotations or FluentAPI

Work with DbContext and DbSet to perform CRUD operations

Create Autocad objects using data queried via EF

Why use an ORM (Object Relational Mapping)?

Why use an ORM (Object Relational Mapping)?

- Databases
  - have their own set of terminology and language (SQL):
    - Tables, Columns, Rows, Indexes, Primary Keys, Foreign Keys, Stored Procedures, Triggers
- Applications
  - strive to model the “problem domain” using the terminology of the application’s users:
    - Projects, Contracts, Equipment
- ORMs bridge the “language” gap by converting from application code to DBMS code

Database First Approach

Visual Designer

EDMX (xml)

TT (text templating)

Existing Database

Model First Approach

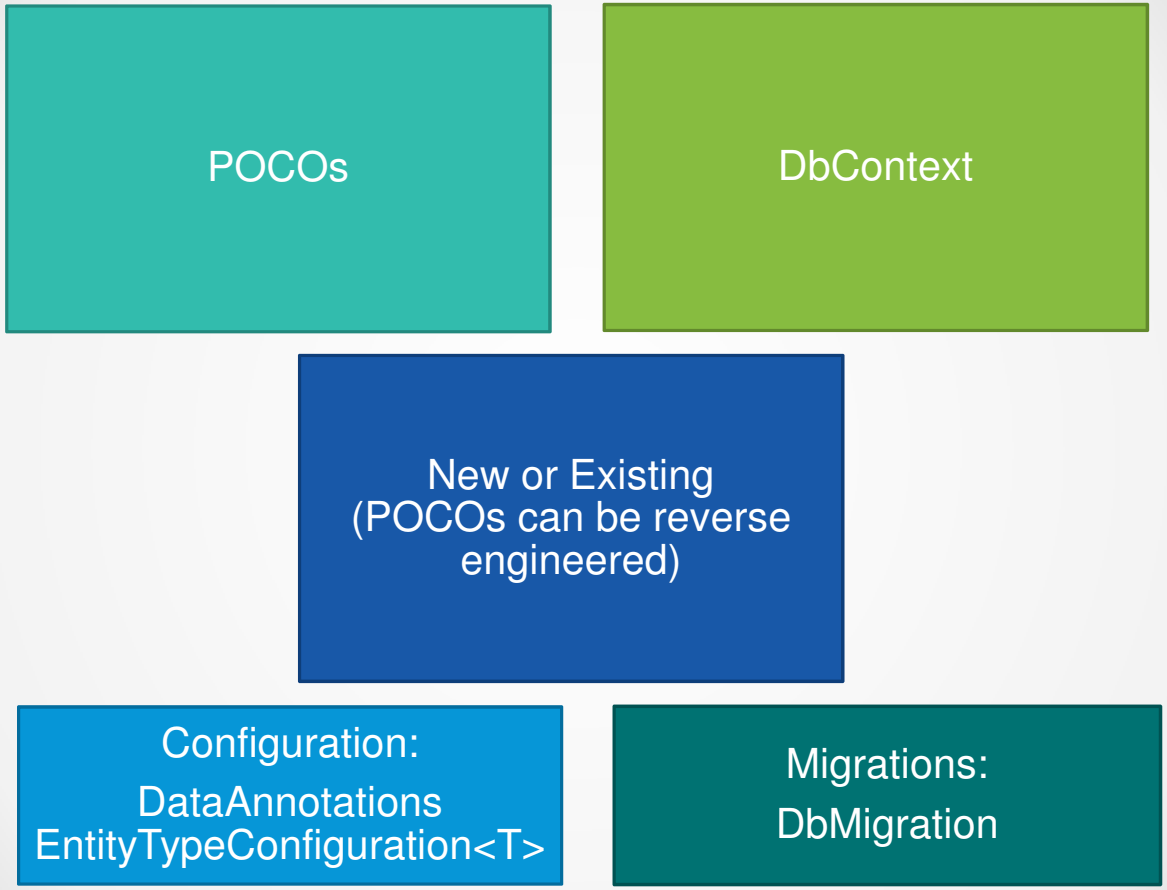
Visual Designer

EDMX (xml)

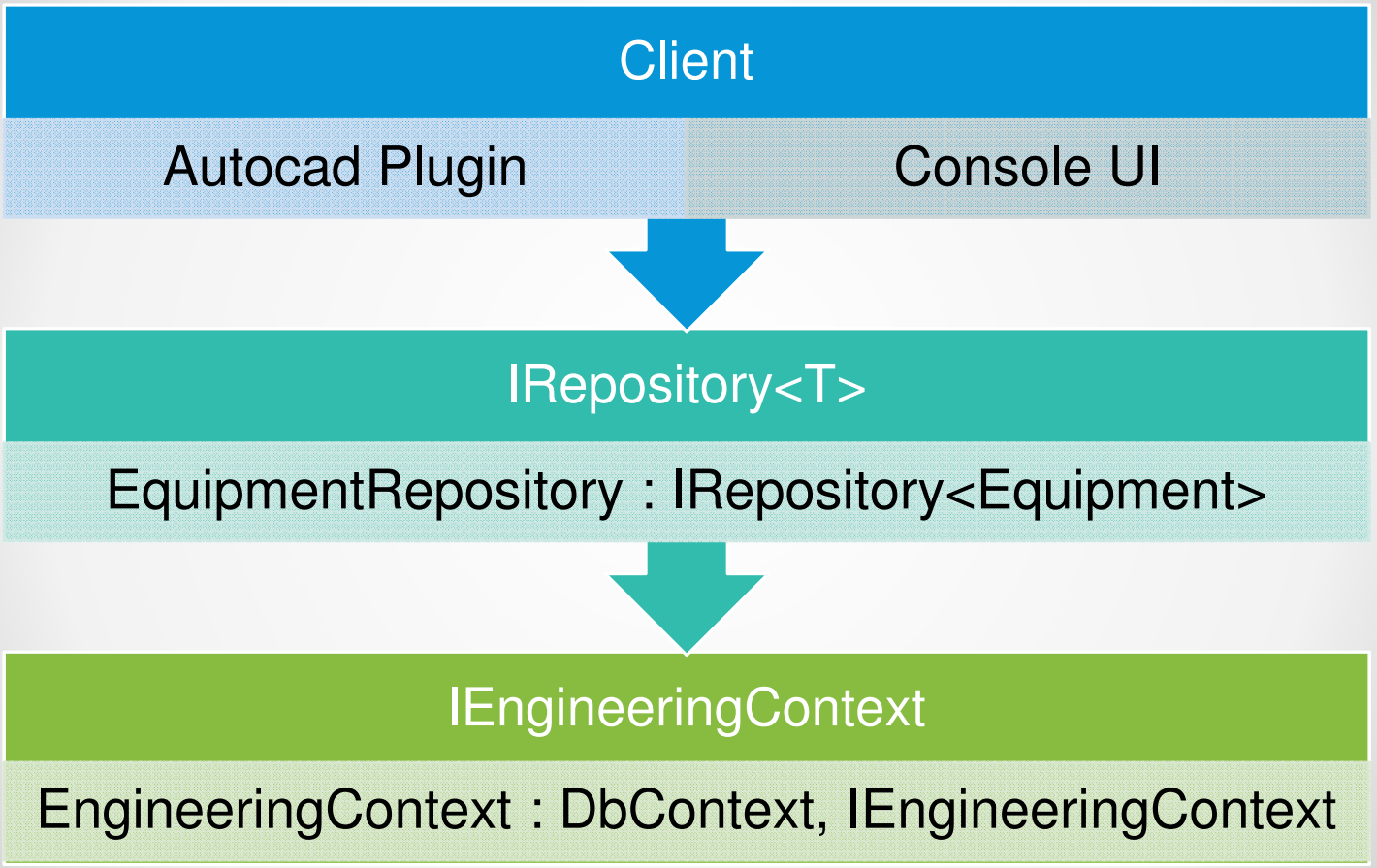
TT (text templating)

New Database

Code First Approach



Abstraction



Multiple Approaches

- Database First
  - Visual Designer, EDMX file
  - Existing database
    - Learning EF
    - Limited access to database(?)
- Model First
  - Visual Designer, EDMX file
  - New database, created from the visual design
- Code First
  - New or existing database
  - Highest control over the code

Key Components of EF

- Model class(es)
  - POCO—plain old class objects that model the application's "problem domain" (ex: Project, Contractor, Equipment)
- Navigation properties and Associations
  - Allow navigation between related objects:
    - var contractorName = project.Contractor.Name;
  - Association identifies how objects are tied together:
    - Primary Key -> Foreign Key relationships in the database (typically)

Key Components of EF

- DbContext
  - Derive from this class, encompasses model and interaction between your model and the database
  - Various constructors to supply connectionString
  - DbSet<T>
    - Represents a set of model objects retrieved from the database (ex: a set of records from the Project table, materialized as Project objects)

Key Components of EF

- DbContext (continued)
  - ChangeTracker
    - Tracks “status” of attached entities (added, modified, deleted)
  - SaveChanges method
    - override to provide “save time” functionality
  - OnModelCreating method
    - Set configuration options for model
    - Works with EntityConfiguration<T> classes
- EntityConfiguration<T>
  - Derive from this for entity specific configuration classes

Key Components of EF

- DbMigrationsConfiguration<DbContext>
  - Configure model options
    - Automatic migrations
    - Data loss allowed
  - Seed method
- DbMigration
  - Generated from model to configure database schema

Migrations

- Creates special upgrade/downgrade classes to modify the database schema
  - DbMigrationsConfiguration derived class
  - Package Manager Console
    - **Enable-migrations** (target project is the one where DbContext lives, only needs to be done once)
    - **add-migration *nameOfMigration*** (use each time you make a configuration change)
    - **update-database** (use after *add-migration*)

Configuration of the Model

- DataAnnotations
  - System.ComponentModel.DataAnnotations
  - Decorate classes and properties with annotations
    - [Table(“Equipment”)]
    - [Required(AllowEmptyStrings=false)]
    - [MaxLength(30)]
- Fluent API
  - Configuration done in code
    - DbContext.OnModelCreating method
    - OR EntityTypeConfiguration<T> derived classes
      - Map(x => x.ToTable(“Contractor”));
      - Property(x => x.Description.HasColumnName(“Desc”).HasMaxLength(300);

Resources

- ***Anything*** by Julie Lerman
  - Pluralsight courses
  - *Programming Entity Framework: Code First*
  - *Programming Entity Framework: DbContext*
  - *Programming Entity Framework: Building Data Centric Apps with ADO.NET Entity Framework*
  - *Data Points* series in MSDN Magazine
- EF Documentation
  - https://msdn.microsoft.com/en-us/data/ee712907
- More in class handout

Class Files

- Class downloads include:
  - *Detailed* explanation/tutorial of what was covered in class
  - Sample projects in VB and C#
  - “Bonus” slides in presentation
- Questions after AU? Please contact me:
  - ben@leadensky.com
  - Twitter: @leadensky
- SD10672: Exploring Entity Framework

Be heard! Provide AU session feedback.

- Via the Survey Stations, email or mobile device.
- AU 2016 passes awarded daily!
- Give your feedback after each session.
- Give instructors feedback in real-time.

Forget to take notes? No problem!

After AU visit:
AutodeskUniversity.com

Click on **My AU** to find:

- Class Recordings
- Presentations
- Handouts

All of your sessions will be there to enjoy again and again.