[MS-DPEDM]:

Entity Data Model Data Portability Overview

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
9/3/2010	0.1	New	Released new document.
2/9/2011	0.2	Minor	Clarified the meaning of the technical content.
7/7/2011	0.2	None	No changes to the meaning, language, or formatting of the technical content.
11/3/2011	0.2	None	No changes to the meaning, language, or formatting of the technical content.
1/19/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
2/23/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
3/27/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
5/24/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
6/29/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	0.2	None	No changes to the meaning, language, or formatting of the technical content.
3/26/2013	0.2	None	No changes to the meaning, language, or formatting of the technical content.
6/11/2013	0.2	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	0.2	None	No changes to the meaning, language, or formatting of the technical content.
12/5/2013	0.2	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2014	0.2	None	No changes to the meaning, language, or formatting of the technical content.
5/20/2014	0.2	None	No changes to the meaning, language, or formatting of the technical content.
5/10/2016	1.0	None	No changes to the meaning, language, or formatting of the technical content.
8/16/2017	1.0	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Entity Data Model Data Portability Overview document provides an overview of data portability for the **Entity Data Model (EDM)**, an **entity**-relationship model that is used to define an application's data model. The data model is represented by using entities and the relationships (that is, **associations**) between those entities.

Three specifications are pertinent to EDM:

- [MC-CSDL]: Conceptual Schema Definition File Format. **Conceptual Schema Definition Language (CSDL)** is a specification language that is used for defining the neutral conceptual mode, which is independent of the storage model used for persisting application data. Because the conceptual model is neutral and independent, application implementers can build a conceptual model in a way that is independent of persistence concerns.
- [MS-SSDL]: Store Schema Definition File Format. **Store Schema Definition Language (SSDL)** is a specification language that is used to formally describe the database **schema** used for persisting the data of an application built on top of EDM.

  The SSDL definition is similar to a CSDL definition; the primary difference between them is that the data types used in SSDL declarations are the actual types supported by a particular database product engine.
- [MS-MSL]: Mapping Specification File Format. **Mapping Specification Language (MSL)** is a specification language that is used to connect the types that are declared in CSDL to store the schema definition that is declared in SSDL.

1.1 Glossary

This document uses the following terms:

association: A named independent relationship between two entity type definitions. Associations in the **Entity Data Model (EDM)** are first-class concepts and are always bidirectional. Indeed, the first-class nature of associations helps distinguish the **EDM** from the relational model. Every association includes exactly two association ends.

conceptual schema definition language (CSDL): A language that is based on XML and that can be used to define conceptual models that are based on the **Entity Data Model (EDM)**.

entity: An instance of an EntityType element that has a unique identity and an independent existence. An entity is an operational unit of consistency.

Entity Data Model (EDM): A set of concepts that describes the structure of data, regardless of its stored form.

mapping specification language (MSL): An XML-based language that can be used to define mapping between a conceptual schema and a store schema.

schema: A container that defines a namespace that describes the scope of EDM types. All EDM types are contained within some namespace.

store schema definition language (SSDL): An XML-based language that can be used to define storage models by using the **Entity Data Model (EDM)**.

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents

in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

[MC-CSDL] Microsoft Corporation, "Conceptual Schema Definition File Format".

[MS-MSL] Microsoft Corporation, "Mapping Specification Language File Format".

[MS-SSDL] Microsoft Corporation, "Store Schema Definition Language File Format".

2 Data Portability Scenarios

2.1 Exposing an Instance of a Data Model by Using CSDL

A third-party product or service can use **CSDL** to expose a data model so that the data model uses the concepts of **EDM** and CSDL to describe the structure and some of the semantics that pertain to the data model. Consumers of this data model can deduce facts about it and use it for various scenarios at a higher level of abstraction without needing to know where the data is physically stored.

Examples of this scenario include the following:

- A customer relationship management (CRM) or enterprise resource planning (ERP) product uses CSDL to describe a data model as a way to express the structure of **entities** and their relationships. Implementers who are building solutions on top of this product can use the schema to deduce facts about this data model without having to concern themselves about its physical schema.
- EDM is used as the **schema** format for enterprise scenarios such as reporting, business intelligence, synchronization, and so on. Building these scenarios on top of EDM eliminates the need to be concerned about the physical store of the data so that the user can focus on the conceptual/domain rules that govern the data model.

2.1.1 Data Description

The data in this scenario is the **CSDL** metadata that is used to represent the data model. **MSL** and **SSDL** are not part of this data portability scenario.

2.1.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols used in this data portability scenario.

Protocol or format name	Description	Reference
Conceptual Schema Definition Language	Metadata definition language that is used to describe the entities and associations in terms of the business/conceptual domain.	[MC-CSDL]

2.1.3 Reverse Engineering a Database and Creating an EDM Model

A **schema** from a relational database, such as Microsoft SQL Server, can be reverse-engineered into an EDM by using the EDMGen.exe tool that is included in the Microsoft .NET Framework 3.5 Service Pack 1 (SP1) and in the Microsoft .NET Framework 4.0. Alternatively, Microsoft Visual Studio 2008 Service Pack 1 (SP1) and Microsoft Visual Studio 2010 can be used.

This data portability scenario describes how to use the EDMGen.exe tool to create an EDM.

To reverse-engineer a database and to create an EDM

1. From a command prompt, run EDMGen.exe by pointing it to the database that you want to reverse engineer into an EDM.

For example, the following command-line command uses the Northwind database schema that is installed on a local instance of Microsoft SQL Server Express. This command reverse engineers an EDM model (**CSDL** file) along with the corresponding **SSDL** schema and **MSL** mapping metadata:

```
edmgen.exe /mode:FullGeneration /connectionString:"server=.\sqlexpress;integrated
security=true;database=Northwind" /project:Northwind /namespace:Northwind
/EntityContainer:NorthwindEntitites
```

2.1.3.1 Preconditions

The database being reverse engineered is required to exist. Additionally, Microsoft Visual Studio 2008, Visual Studio 2010, and EDMGen.exe require an Entity Framework–enabled ADO.NET provider for connectivity to the database. Database product vendors and third parties offer database providers that enable ADO.NET-based connectivity to relational databases.

2.1.3.2 Versioning

Versioning considerations for **CSDL**, **SSDL**, and **MSL** are documented in [MC-CSDL], [MS-SSDL], and [MS-MSL], respectively.

2.1.3.3 Error Handling

None.

2.1.3.4 Coherency Requirements

There are no special coherency requirements.

2.1.3.5 Additional Considerations

There are no additional considerations.

2.2 Implementing Visual Designers and Tooling for CSDL, SSDL, and MSL

A third party can build a visual design tool that enables its users to build customized, EDM-based models. The tool can support various aspects, such as reverse engineering an existing database, that enable users to create a new model or support various code generation strategies based on the model that is being created.

2.2.1 Data Description

The data in this scenario consists of the **CSDL**, **SSDL**, and **MSL** metadata that is generated by the design tool.

2.2.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols used in this data portability scenario.

Protocol or format name	Description	References
Conceptual Schema Definition Language	Metadata definition language that is used to describe the entities and associations in terms of the business/conceptual domain.	[MC-CSDL]
Store Schema Definition Language	Metadata definition language that is used to describe the physical store schema used for persisting the data in a relational database.	[MS-SSDL]

Protocol or format name	Description	References
Mapping Specification Language	Mapping language that is used to map the concepts in the store schema to the concepts in the conceptual schema.	[MS-MSL]

2.2.3 Generating and Persisting an EDM–Based Metadata to Disk

In this scenario, the design tool is responsible for the following functionalities:

- Enabling users to build **CSDL**, **SSDL**, and **MSL** metadata.

  The tool is to adhere to the syntactic and semantic rules for generating CSDL, SSDL, and MSL metadata so that the **schema** can then be used and deployed in a variety of scenarios.

- Enabling the data portability scenario described in section 2.1.3

- Performing the functions that are done by the EDM designer in Visual Studio or EDMGen.exe.

There are no prescriptive requirements for how a design tool connects to the database, generates metadata, or persists XML for the metadata on disk.

The rules for generating metadata that are described in [MC-CSDL], [MS-SSDL], and [MS-MSL] are to be followed.

2.2.3.1 Preconditions

None.

2.2.3.2 Versioning

Versioning considerations for **CSDL**, **SSDL**, and **MSL** are documented in [MC-CSDL], [MS-SSDL], and [MS-MSL], respectively.

2.2.3.3 Error Handling

None.

2.2.3.4 Coherency Requirements

There are no special coherency requirements.

2.2.3.5 Additional Considerations

There are no additional considerations.

3 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

4 Index