

ⓘ We're no longer updating this content regularly. Check the [Microsoft Product Lifecycle](#) for information about how this product, service, technology, or API is supported.

Return to main site

✕

🔍 Filter by title

WCF Data Services 4.5

Overview

> Getting Started

> Defining WCF Data Services

▼ WCF Data Services Client Library

WCF Data Services Client Library

▼ Generating the Data Service Client Library

**Generating the Data Service Client Library**

WCF Data Service Client Utility (DataSvcUtil.exe)

How to: Add a Data Service Reference

How to: Manually Generate Client Data Service Classes

> Querying the Data Service

> Loading Deferred Content

> Updating the Data Service

> Asynchronous Operations

> Batching Operations

> Binding Data to Controls

Calling Service Operations

Managing the Data Service Context

Working with Binary Data

⋯ / [Previous versions](#) / [.NET](#) / [.NET Framework](#) / [WCF Data Services](#) / ⊕ ⋮

# Generating the Data Service Client Library (WCF Data Services)

Article • 03/30/2017 • [2 contributors](#)

## In this article

- [Client Data Type Mapping](#)
- [See also](#)

### ⓘ Important

WCF Data Services has been deprecated and will no longer be available for download from the Microsoft Download Center. WCF Data Services supported earlier versions of the Microsoft OData (V1-V3) protocol only and has not been under active development. OData V1-V3 has been superseded by OData V4, which is an industry standard published by OASIS and ratified by ISO. OData V4 is supported through the OData V4 compliant core libraries available at [Microsoft.OData.Core](#) [↗](#). Support documentation is available at [OData.Net](#) [↗](#), and the OData V4 service libraries are available at [Microsoft.AspNetCore.OData](#) [↗](#).

[RESTier](#) [↗](#) is the successor to WCF Data Services. RESTier helps you bootstrap a standardized, queryable, HTTP-based REST interface in minutes. Like WCF Data Services before it, Restier provides simple and straightforward ways to shape queries and intercept submissions before and after they hit the database. And like Web API + OData, you still have the flexibility to add your own custom queries and actions with techniques you're already familiar with.

A data service that implements the Open Data Protocol (OData) can return a service metadata document that describes the data model exposed by the OData feed. For more information, see the Service Metadata Document section in the [OData: Overview](#) [↗](#) article. You can use the **Add Service Reference** dialog in Visual Studio to add a reference to an OData-based service. When you use this tool to add a reference to the metadata returned by an OData feed in a client project, it performs the following actions:

- Requests the service metadata document from the data service and interprets the returned metadata.

### ⓘ Note

The returned metadata is stored in the client project as an .edmx file. This .edmx file cannot be opened by using the Entity Data Model designer because it does not have the same format an .edmx file used by the Entity Framework. You can view this metadata file by using the XML editor or any text editor. For more information, see [\[MC-EDMX\]: Entity Data Model for Data Services Packaging Format](#).

- Generates a representation of the service as an entity container class that inherits from [DataServiceContext](#). This generated entity container class resembles the entity container

that the Entity Data Model tools generate. For more information, see [Object Services Overview \(Entity Framework\)](#).

- Generates data classes for the data model types that it discovers in the service metadata.
- Adds a reference to the `System.Data.Services.Client` assembly to the project.

For more information, see [How to: Add a Data Service Reference](#).

The client data service classes can also be generated by using the `DataSvcUtil.exe` tool at the command prompt. For more information, see [How to: Manually Generate Client Data Service Classes](#).

## Client Data Type Mapping

When you use the **Add Service Reference** dialog in Visual Studio or the `DataSvcUtil.exe` tool to generate client data classes that are based on an OData feed, the .NET Framework data types are mapped to the primitive types from the data model as follows:

 Expand table

Data model type	.NET Framework data type
<code>Edm.Binary</code>	<code>Byte []</code>
<code>Edm.Boolean</code>	<code>Boolean</code>
<code>Edm.Byte</code>	<code>Byte</code>
<code>Edm.DateTime</code>	<code>DateTime</code>
<code>Edm.Decimal</code>	<code>Decimal</code>
<code>Edm.Double</code>	<code>Double</code>
<code>Edm.Guid</code>	<code>Guid</code>
<code>Edm.Int16</code>	<code>Int16</code>
<code>Edm.Int32</code>	<code>Int32</code>
<code>Edm.Int64</code>	<code>Int64</code>
<code>Edm.SByte</code>	<code>SByte</code>
<code>Edm.Single</code>	<code>Single</code>
<code>Edm.String</code>	<code>String</code>

For more information, see the Primitive Data Types section in the [OData: Overview](#) article.

## See also

- [WCF Data Services Client Library](#)
- [Quickstart](#)