[MS-DPMDS]:

Master Data Services Data Portability Overview

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may 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, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

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

Table of Contents

1 Introduction

The Master Data Services Data Portability Overview document provides an overview of data portability for Master Data Services (MDS), a Master Data Management (MDM) application that is built from platform components. MDS can be deployed as an application or can be extended by using these platform components to consistently define and manage the critical data **entities** of an organization.

MDS helps organizations standardize and streamline the business data that customers use to make critical business decisions. It is an any-domain hub that supports, but is not limited to, domains such as product, customer, location, cost center, equipment, employee, and vendor.

By using MDS, organizations can manage critical data assets by enabling proactive stewardship, enforcing data quality rules, defining workflows around data changes, notifying impacted parties, managing hierarchies, and sharing the authoritative source with all impacted systems.

1.1 Glossary

This document uses the following terms:

attribute group: A collection of attributes that can be used to decorate an XML element, as described in [XMLSCHEMA1].

business rule: A user-defined process that can proactively manage data within the Master Data Services (MDS) database.

collection: A user-defined group of data items from the same **entity**.

entity: Tabular data that is stored within the Master Data Services (MDS) system.

hierarchy: An arrangement of data items within the Master Data Services (MDS) system in a ranked or graduated series.

model: The highest level of data organization in Master Data Services. A model contains objects and entities.

model deployment package: The XML file that contains the MDS **model** metadata (schema) and that can optionally contain the user's **model** data (master data).

Model Deployment Wizard: The wizard UI in MDS that is used by the user to create a **model deployment package** or to deploy a previously created **model deployment package**.

Windows Communication Foundation (WCF): A framework for building connected service-oriented applications.

XML: The Extensible Markup Language, as described in [XML1.0].

XML schema definition (XSD): The World Wide Web Consortium (W3C) standard language that is used in defining XML schemas. Schemas are useful for enforcing structure and constraining the types of data that can be used validly within other XML documents. XML schema definition refers to the fully specified and currently recommended standard for use in authoring XML schemas.

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.

[MS-SSMDSWS-15] Microsoft Corporation, "Master Data Services Web Service 15".

[MS-SSMDSWS] Microsoft Corporation, "Master Data Services Web Service".

[MSDN-MDM] Microsoft Corporation, "Model Deployment Wizard (Master Data Manager, System Administration)", http://msdn.microsoft.com/en-us/library/ee633778(v=SQL.105).aspx

[MSDN-MDPXS] Microsoft Corporation, "Model Deployment Package XML Schema (Master Data Services)", http://msdn.microsoft.com/en-us/library/ff714018.aspx

[XML10/5] Bray, T., Paoli, J., Sperberg-McQueen, C.M., et al., Eds., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, http://www.w3.org/TR/2008/REC-xml-20081126/

2 Data Portability Scenarios

2.1 Create and Deploy a Deployment Package

An architect who is building a new **model** builds the model in the organization's MDS development system. He then uses the MDS model deployment feature to package the model—including its entities, **hierarchies**, **collections**, **attribute group**s, and **business rules**—into a deployment package, and then saves it to a network share.

The architect then starts the administrator UI for the organization's MDS quality assurance (QA) environment. In the QA MDS system, he launches the model deployment feature, navigates to the customer deployment package that he created on the development system, and then deploys it in the QA system. The model and all of its contents are deployed in the QA system and are ready for testing by the organization's customer data stewards.

After testing and sign off by the data stewards, the architect now must deploy the customer model to the production MDS system. He logs on to the development system, starts the model deployment feature, creates a new deployment package for the customer model, and then saves it to a network share. He then logs on to the production system, starts the model deployment feature, and then navigates to the customer deployment package that he created on the development system. With a few clicks, he deploys the new customer model on the production system.

2.1.1 Data Description

The model deployment process can include data within the **model deployment package**. For the purposes of this document, this data is the user data. This user data is commonly known as master data.

Master data consists of the critical nouns of a business and falls into four groupings: people, things, places, and concepts. Further categorizations within those groupings are called subject areas, domain areas, or entity types.

For example, within the people grouping, the domain areas may include customer, employee, and salesperson. Within the things grouping, the domain areas may include product, part, store, and asset. Within the concepts grouping, the domain areas may include contract, warrantee, and licenses. Finally, within the places grouping, the domain areas may include office locations and geographic divisions.

Some of these domain areas may be further divided. For example, the customer domain area may be further divided based on incentives and history, or a company may have regular customers as well as premiere and executive customers. The product domain area may be further divided by sector and industry. For example, the requirements, life cycle, and create, read, update, and delete cycle for a product in the consumer packaged goods (CPG) sector are likely to be very different from those of products in the clothing industry.

The granularity of domains is determined by the magnitude of differences between the attributes of the entities within the domains.

2.1.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols that are used in the data portability scenarios.

Protocol or format name	Description	Reference
MDS	Master Data Services	[MS-SSMDSWS]

Protocol or format name	Description	Reference
		[MS-SSMDSWS-15]
XML	Extensible Markup Language	[XML10/5]
Model Deployment XSD	Model Deployment Format XSD	[MSDN-MDPXS]

2.1.3 Data Portability Methodology

Export

The process for full model data extraction from MDS is done through the **Model Deployment Wizard** UI [MSDN-MDM]. The wizard simplifies the task of selecting the model and data, including the data version, for extraction into the package file. The following steps are performed to create the model deployment package:

1. After launching the wizard from the **System Administration** page in MDS, the user clicks **Create**. From a drop-down list that contains all models in the MDS system, the user selects and the model to use to create the package file. The drop-down list is truncated based on the user's permissions settings.

2. The user selects whether to include model data in the package and the data version from which to extract that data. When the user clicks **Next** in the wizard, the wizard internally calls the appropriate APIs to extract the model metadata and data and to serialize it into **XML** by using native serialization in **Windows Communication Foundation (WCF)**.

3. After the package is created, in the wizard, the user specifies the location in which to save the package file. Saving the package file completes the package creation process.

The XML in the file conforms to the **XSD** that is defined in [MSDN-MDPXS]. The file format of the package is defined in Appendix A: Model Deployment XSD, section 3.

Using the model deployment API is similar to using the UI. The user calls the Create API, and then specifies the model, indicates whether to include data and the data version (if data is selected), and then specifies the location for the package file. The API handles the extraction process in the same way that the UI does. The details of the extraction are opaque to the user. The user specifies the model, whether to include data, and the data version.

2.1.3.1 Preconditions

To create the package by using the wizard, the user must have administrator permissions on the source system. To deploy the package, the user must have administrator permissions on the target system.

2.1.3.2 Versioning

This data portability scenario has no special versioning requirements. The data is versioned within MDS; if the user decides to include data in the package, the user selects the data version to include.

2.1.3.3 Error Handling

If business rules exist that rely on data values and if data is not included in the package, the deployment fails.

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 Deploy a Prepackaged ISV Deployment Package

A consulting company has created an MDS product master model that is specifically geared to the pharmaceutical industry. A pharmaceutical corporation has engaged the consulting company to implement the product model on its MDS landscape. A consultant brings the product model to the pharmaceutical corporation on her USB flash drive.

Working with a data architect and a database administrator, the consultant copies the model to a network share on the pharmaceutical corporation's network. The database administrator logs on to the MDS development system and starts the Model Deployment feature. With a few clicks, he deploys the product model on the MDS development system. The architect can then begin to evaluate the utility of the prepackaged product model provided by the consultant.

2.2.1 Data Description

The model deployment process can include data within the model deployment package. For the purposes of this document, this data is the user data. This user data is commonly known as master data.

Master data consists of the critical nouns of a business and falls generally into four groupings: people, things, places, and concepts. Further categorizations within those groupings are called subject areas, domain areas, or entity types.

For example, within the people grouping, the domain areas may include customer, employee, and salesperson. Within the things grouping, the domain areas may include product, part, store, and asset. Within the concepts grouping, the domain areas may include contract, warrantee, and licenses. Finally, within the places grouping, the domain areas may include office locations and geographic divisions.

Some of these domain areas may be further divided. For example, the customer domain area may be further divided based on incentives and history, or a company may have regular customers in addition to premiere and executive customers. The product domain area may be further divided by sector and industry. For example, the requirements, life cycle, and create, read, update, and delete cycle for a product in the CPG sector are likely to be different from those of products in the clothing industry.

The granularity of domains is determined by the magnitude of differences between the attributes of the entities within the domains.

2.2.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols that are used in the data portability scenarios.

Protocol or format name	Description	Reference
MDS	Master Data Services	[MS-SSMDSWS] [MS-SSMDSWS-15]
XML	Extensible Markup Language	[XML10/5]

Protocol or format name	Description	Reference
Model Deployment XSD	Model Deployment Format XSD	[MSDN-MDPXS]

2.2.3 Data Portability Methodology

Import

The process for importing a model deployment package into an MDS system is done through the Model Deployment Wizard UI [MSDN-MDM]. The wizard simplifies the task of deploying the package file. The following steps are performed to deploy the model deployment package:

1. After launching the wizard from the **System Administration** page in MDS, the user clicks **Deploy**, and then browses to the package file to deploy.
2. The package is uploaded and validated when the user clicks **Next**. If the package is successfully validated, the deployment process starts.
    - If the model that is contained in the package does not exist on the target system, the deployment process runs. The wizard handles the deployment process by calling the appropriate APIs. The details of the extraction are opaque to the user; an animated wait dialog is displayed during the extraction.
    - If the model that is contained in the package does exist on the target system, the user is presented with two choices. The user can apply the content of the package as an update of the model on the target or can choose to create a new model on the target by using the contents of the package. After this selection is made, the wizard handles the deployment process by calling the appropriate APIs. The details of the extraction are opaque to the user; an animated wait dialog is displayed during the extraction.
3. The process finishes. The user exits the dialog after the deployment is successful.

The XML in the file conforms to the XSD that is defined in [MSDN-MDPXS]. The file format of the package is defined in Appendix A: Model Deployment XSD, section 3.

Using the model deployment API is similar to using the UI. The user calls the Deploy API, and then specifies the location of the package file. The API handles the deployment process in the same way that the UI does. The details of the deployment are opaque to the user.

2.2.3.1 Preconditions

To create the package by using the wizard, the user must have administrator permissions on the source system. To deploy the package, the user must have administrator permissions on the target system.

2.2.3.2 Versioning

This data portability scenario has no special versioning requirements. The data is versioned within MDS; if the user decides to include data in the package, the user selects the data version to include.

2.2.3.3 Error Handling

There are no special error handling requirements.

2.2.3.4 Coherency Requirements

There are no special coherency requirements.

2.2.3.5 Additional Considerations

There are no additional considerations.

3 Appendix A: Model Deployment XSD

Microsoft SQL Server version	Section
Microsoft SQL Server 2008 R2 Microsoft SQL Server 2012 Microsoft SQL Server 2014	3.1
Microsoft SQL Server 2016	3.2

3.1 Model Deployment XSD for SQL Server 2008 R2, SQL Server 2012, and SQL Server 2014

3.1.1 Elements

The following XSD shows the individual elements for a package.

```
<?xml version="1.0" encoding="utf-8"?> <xs:schema
xmlns:a="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
xmlns:b="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
xmlns:i="http://www.w3.org/2001/XMLSchema-instance" attributeFormDefault="unqualified"
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.MasterDataServices.Deploym
ent" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09" />
  <xs:element name="Package">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="businessRuleSet">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="a:BRActions" />
              <xs:element ref="a:BRConditionTreeNodes" />
              <xs:element ref="a:BRConditions" />
              <xs:element ref="a:BusinessRules" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="masterData" nillable="true" />
        <xs:element name="metadata">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="a:AttributeGroups" />
              <xs:element ref="a:Attributes" />
              <xs:element ref="a:DerivedHierarchies" />
              <xs:element ref="a:DerivedHierarchyLevels" />
              <xs:element ref="a:Entities" />
              <xs:element ref="a:ExplicitHierarchies" />
              <xs:element ref="a:MemberTypes" />
              <xs:element ref="a:Models" />
              <xs:element ref="a:VersionFlags" />
              <xs:element ref="a:Versions" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="settings">
          <xs:complexType>
            <xs:sequence>
```

```
              <xs:element name="assemblyVersion" type="xs:string" />
              <xs:element name="containsData" type="xs:boolean" />
              <xs:element name="createdBy" type="xs:string" />
              <xs:element name="createdDate" type="xs:dateTime" />
              <xs:element name="deploymentType" type="xs:string" />
              <xs:element name="productVersion" type="xs:string" />
              <xs:element name="version" type="xs:unsignedByte" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

3.1.2 Arrays

The following XSD shows how to serialize arrays for data elements that have multiple lines of data.

```
<?xml version="1.0" encoding="utf-8"?>
 <xs:schema xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
attributeFormDefault="unqualified" elementFormDefault="qualified"
 targetNamespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
 xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09" />
  <xs:element name="anyType">
    <xs:complexType>
      <xs:sequence>
        <xs:element
xmlns:q1="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
ref="q1:Identifier" />
        <xs:element
xmlns:q2="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
ref="q2:PropertyName" />
        <xs:element minOccurs="0"
xmlns:q3="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
ref="q3:AttributeCode" />
        <xs:element minOccurs="0"
xmlns:q4="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09" ref="q4:Value"
/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

3.1.3 Package Structure

The following XSD shows the full package structure.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
attributeFormDefault="unqualified" elementFormDefault="qualified"
targetNamespace="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/2003/10/Serialization/Arrays" />
  <xs:element name="BRActions" />
  <xs:element name="BRConditionTreeNodes" />
  <xs:element name="BRConditions" />
  <xs:element name="BusinessRules">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="BusinessRule">
```

```
            <xs:complexType>
              <xs:sequence>
                <xs:element name="AuditInfo" nillable="true" />
                <xs:element name="BRActions">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element maxOccurs="unbounded" name="BRAction">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="AuditInfo" nillable="true" />
                            <xs:element name="BusinessRuleId">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="Id" type="xs:string" />
                                  <xs:element name="Name" type="xs:string" />
                                  <xs:element name="InternalId" type="xs:unsignedByte" />
                                  <xs:element name="ModelId">
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element name="Id" type="xs:string" />
                                        <xs:element name="Name" type="xs:string" />
                                        <xs:element name="InternalId" type="xs:unsignedByte" />
                                      </xs:sequence>
                                    </xs:complexType>
                                  </xs:element>
                                  <xs:element name="EntityId">
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element name="Id" type="xs:string" />
                                        <xs:element name="Name" type="xs:string" />
                                        <xs:element name="InternalId" type="xs:unsignedByte" />
                                      </xs:sequence>
                                    </xs:complexType>
                                  </xs:element>
                                  <xs:element name="MemberType" type="xs:string" />
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element name="Identifier">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="Id" type="xs:string" />
                                  <xs:element name="Name" nillable="true" />
                                  <xs:element name="InternalId" type="xs:unsignedByte" />
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element name="Operator" type="xs:string" />
                            <xs:element name="PostfixArguments">
                              <xs:complexType>
                                <xs:sequence minOccurs="0">
                                  <xs:element maxOccurs="unbounded"
xmlns:q1="http://schemas.microsoft.com/2003/10/Serialization/Arrays" ref="q1:anyType" />
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                            <xs:element name="Sequence" type="xs:unsignedByte" />
                            <xs:element name="Text" type="xs:string" />
                            <xs:element name="PrefixArgument">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="Identifier">
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element name="Id" type="xs:string" />
                                        <xs:element name="Name" nillable="true" />
                                        <xs:element name="InternalId" type="xs:unsignedByte" />
                                      </xs:sequence>
                                    </xs:complexType>
```

```
                                </xs:element>
                                <xs:element name="PropertyName" type="xs:string" />
                                <xs:element name="AttributeId">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Id" type="xs:string" />
                                      <xs:element name="Name" type="xs:string" />
                                      <xs:element name="InternalId" type="xs:unsignedShort"
/>
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="ChildArguments" />
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="BRConditionTree" nillable="true">
                <xs:complexType>
                  <xs:sequence minOccurs="0">
                    <xs:element name="AuditInfo" nillable="true" />
                    <xs:element name="BRConditions">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="BRCondition">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="AuditInfo" nillable="true" />
                                <xs:element name="BusinessRuleId">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Id" type="xs:string" />
                                      <xs:element name="Name" type="xs:string" />
                                      <xs:element name="InternalId" type="xs:unsignedByte" />
                                      <xs:element name="ModelId">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" type="xs:string" />
                                            <xs:element name="InternalId"
type="xs:unsignedByte" />
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="EntityId">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" type="xs:string" />
                                            <xs:element name="InternalId"
type="xs:unsignedByte" />
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="MemberType" type="xs:string" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="Identifier">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Id" type="xs:string" />
                                      <xs:element name="Name" nillable="true" />
                                      <xs:element name="InternalId" type="xs:unsignedByte" />
```

```
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="Operator" type="xs:string" />
                                <xs:element name="PostfixArguments">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element
xmlns:q2="http://schemas.microsoft.com/2003/10/Serialization/Arrays" ref="q2:anyType" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="Sequence" type="xs:unsignedByte" />
                                <xs:element name="Text" type="xs:string" />
                                <xs:element name="ConditionTreeNodeId">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Id" type="xs:string" />
                                      <xs:element name="Name" nillable="true" />
                                      <xs:element name="InternalId" type="xs:unsignedByte" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                                <xs:element name="PrefixArgument">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Identifier">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" nillable="true" />
                                            <xs:element name="InternalId"
type="xs:unsignedByte" />
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="PropertyName" type="xs:string" />
                                      <xs:element name="AttributeId">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" type="xs:string" />
                                            <xs:element name="InternalId"
type="xs:unsignedShort" />
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="ChildArguments" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="BusinessRuleId">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Id" type="xs:string" />
                          <xs:element name="Name" type="xs:string" />
                          <xs:element name="InternalId" type="xs:unsignedByte" />
                          <xs:element name="ModelId">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Id" type="xs:string" />
                                <xs:element name="Name" type="xs:string" />
                                <xs:element name="InternalId" type="xs:unsignedByte" />
```

```
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="EntityId">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Id" type="xs:string" />
                  <xs:element name="Name" type="xs:string" />
                  <xs:element name="InternalId" type="xs:unsignedByte" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="MemberType" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="ConditionTreeChildNodes" />
      <xs:element name="ConditionTreeParentNode">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Id" type="xs:string" />
            <xs:element name="Name" nillable="true" />
            <xs:element name="InternalId" type="xs:unsignedByte" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="Identifier">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Id" type="xs:string" />
            <xs:element name="Name" nillable="true" />
            <xs:element name="InternalId" type="xs:unsignedByte" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="LogicalOperator" type="xs:string" />
      <xs:element name="Sequence" type="xs:unsignedByte" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Description" type="xs:string" />
<xs:element name="Identifier">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Id" type="xs:string" />
      <xs:element name="Name" type="xs:string" />
      <xs:element name="InternalId" type="xs:unsignedByte" />
      <xs:element name="ModelId">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Id" type="xs:string" />
            <xs:element name="Name" type="xs:string" />
            <xs:element name="InternalId" type="xs:unsignedByte" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="EntityId">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Id" type="xs:string" />
            <xs:element name="Name" type="xs:string" />
            <xs:element name="InternalId" type="xs:unsignedByte" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="MemberType" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
            <xs:element name="NotificationGroup" nillable="true" />
            <xs:element name="NotificationUser" nillable="true" />
            <xs:element name="Priority" type="xs:unsignedByte" />
            <xs:element name="RuleActionText" type="xs:string" />
            <xs:element name="RuleConditionText" type="xs:string" />
            <xs:element name="Status" type="xs:string" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Identifier">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Id" type="xs:string" />
      <xs:element name="Name" nillable="true" />
      <xs:element name="InternalId" type="xs:unsignedByte" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="PropertyName" type="xs:string" />
<xs:element name="Value" type="xs:decimal" />
<xs:element name="AttributeCode" type="xs:string" />
<xs:element name="AttributeGroups" />
<xs:element name="Attributes" />
<xs:element name="DerivedHierarchies" />
<xs:element name="DerivedHierarchyLevels" />
<xs:element name="Entities" />
<xs:element name="ExplicitHierarchies" />
<xs:element name="MemberTypes" />
<xs:element name="Models">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Model">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Identifier">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Id" type="xs:string" />
                  <xs:element name="Name" type="xs:string" />
                  <xs:element name="InternalId" type="xs:unsignedByte" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Permission" type="xs:string" />
            <xs:element name="DerivedHierarchies">
              <xs:complexType>
                <xs:sequence>
                  <xs:element maxOccurs="unbounded" name="DerivedHierarchy">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Identifier">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:element name="Id" type="xs:string" />
                              <xs:element name="Name" type="xs:string" />
                              <xs:element name="InternalId" type="xs:unsignedByte" />
                              <xs:element name="ModelId">
                                <xs:complexType>
                                  <xs:sequence>
                                    <xs:element name="Id" type="xs:string" />
                                    <xs:element name="Name" type="xs:string" />
                                    <xs:element name="InternalId" type="xs:unsignedByte" />
                                  </xs:sequence>
                                </xs:complexType>
                              </xs:element>
                            </xs:sequence>
```

```
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="Permission" type="xs:string" />
                          <xs:element name="AnchorNullRecursions" type="xs:boolean" />
                          <xs:element name="FullyQualifiedName" type="xs:string" />
                          <xs:element name="IsRecursive" type="xs:boolean" />
                          <xs:element name="Levels">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element maxOccurs="unbounded"
name="DerivedHierarchyLevel">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Identifier">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" type="xs:string" />
                                            <xs:element name="InternalId"
type="xs:unsignedByte" />
                                            <xs:element name="ModelId">
                                              <xs:complexType>
                                                <xs:sequence>
                                                  <xs:element name="Id" type="xs:string" />
                                                  <xs:element name="Name" type="xs:string" />
                                                  <xs:element name="InternalId"
type="xs:unsignedByte" />
                                                </xs:sequence>
                                              </xs:complexType>
                                            </xs:element>
                                            <xs:element name="DerivedHierarchyId">
                                              <xs:complexType>
                                                <xs:sequence>
                                                  <xs:element name="Id" type="xs:string" />
                                                  <xs:element name="Name" type="xs:string" />
                                                  <xs:element name="InternalId"
type="xs:unsignedByte" />
                                                </xs:sequence>
                                              </xs:complexType>
                                            </xs:element>
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="Permission" type="xs:string" />
                                      <xs:element name="DisplayName" type="xs:string" />
                                      <xs:element name="ForeignEntityId">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" type="xs:string" />
                                            <xs:element name="InternalId"
type="xs:unsignedByte" />
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="ForeignId">
                                        <xs:complexType>
                                          <xs:sequence>
                                            <xs:element name="Id" type="xs:string" />
                                            <xs:element name="Name" type="xs:string" />
                                            <xs:element name="InternalId"
type="xs:unsignedShort" />
                                          </xs:sequence>
                                        </xs:complexType>
                                      </xs:element>
                                      <xs:element name="ForeignType" type="xs:string" />
                                      <xs:element name="IsRecursive" type="xs:boolean" />
                                      <xs:element name="IsVisible" type="xs:boolean" />
```

```
                                    <xs:element name="LevelNumber" type="xs:unsignedByte"
/>
                                    <xs:element name="MemberType" type="xs:string" />
                                  </xs:sequence>
                                </xs:complexType>
                              </xs:element>
                            </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Entities">
              <xs:complexType>
                <xs:sequence>
                  <xs:element maxOccurs="unbounded" name="Entity">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Identifier">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:element name="Id" type="xs:string" />
                              <xs:element name="Name" type="xs:string" />
                              <xs:element name="InternalId" type="xs:unsignedByte" />
                              <xs:element name="ModelId">
                                <xs:complexType>
                                  <xs:sequence>
                                    <xs:element name="Id" type="xs:string" />
                                    <xs:element name="Name" type="xs:string" />
                                    <xs:element name="InternalId" type="xs:unsignedByte" />
                                  </xs:sequence>
                                </xs:complexType>
                              </xs:element>
                            </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                        <xs:element name="Permission" type="xs:string" />
                        <xs:element name="ExplicitHierarchies">
                          <xs:complexType>
                            <xs:sequence minOccurs="0">
                              <xs:element maxOccurs="unbounded" name="ExplicitHierarchy">
                                <xs:complexType>
                                  <xs:sequence>
                                    <xs:element name="Identifier">
                                      <xs:complexType>
                                        <xs:sequence>
                                          <xs:element name="Id" type="xs:string" />
                                          <xs:element name="Name" type="xs:string" />
                                          <xs:element name="InternalId"
type="xs:unsignedByte" />
                                          <xs:element name="ModelId">
                                            <xs:complexType>
                                              <xs:sequence>
                                                <xs:element name="Id" type="xs:string" />
                                                <xs:element name="Name" type="xs:string" />
                                                <xs:element name="InternalId"
type="xs:unsignedByte" />
                                              </xs:sequence>
                                            </xs:complexType>
                                          </xs:element>
                                          <xs:element name="EntityId">
                                            <xs:complexType>
                                              <xs:sequence>
                                                <xs:element name="Id" type="xs:string" />
                                                <xs:element name="Name" type="xs:string" />
```

```
                                                <xs:element name="InternalId"
type="xs:unsignedByte" />
                                              </xs:sequence>
                                            </xs:complexType>
                                          </xs:element>
                                        </xs:sequence>
                                      </xs:complexType>
                                    </xs:element>
                                    <xs:element name="Permission" type="xs:string" />
                                    <xs:element name="FullyQualifiedName" type="xs:string"
/>
                                    <xs:element name="IsMandatory" type="xs:boolean" />
                                  </xs:sequence>
                                </xs:complexType>
                              </xs:element>
                            </xs:sequence>
                          </xs:complexType>
                        </xs:element>
                        <xs:element name="IsBase" type="xs:boolean" />
                        <xs:element name="IsFlat" type="xs:boolean" />
                        <xs:element name="IsSystem" type="xs:boolean" />
                        <xs:element name="MemberTypes">
                          <xs:complexType>
                            <xs:sequence>
                              <xs:element maxOccurs="unbounded" name="EntityMemberType">
                                <xs:complexType>
                                  <xs:sequence>
                                    <xs:element name="Identifier">
                                      <xs:complexType>
                                        <xs:sequence>
                                          <xs:element name="Id" type="xs:string" />
                                          <xs:element name="Name" nillable="true" />
                                          <xs:element name="InternalId"
type="xs:unsignedByte" />
                                        </xs:sequence>
                                      </xs:complexType>
                                    </xs:element>
                                    <xs:element name="Permission" type="xs:string" />
                                    <xs:element name="AttributeGroups">
                                      <xs:complexType>
                                        <xs:sequence minOccurs="0">
                                          <xs:element maxOccurs="unbounded"
name="AttributeGroup">
                                            <xs:complexType>
                                              <xs:sequence>
                                                <xs:element name="Identifier">
                                                  <xs:complexType>
                                                    <xs:sequence>
                                                      <xs:element name="Id"
type="xs:string" />
                                                      <xs:element name="Name"
type="xs:string" />
                                                      <xs:element name="InternalId"
type="xs:unsignedByte" />
                                                      <xs:element name="ModelId">
                                                        <xs:complexType>
                                                          <xs:sequence>
                                                            <xs:element name="Id"
type="xs:string" />
                                                            <xs:element name="Name"
type="xs:string" />
                                                            <xs:element name="InternalId"
type="xs:unsignedByte" />
                                                          </xs:sequence>
                                                        </xs:complexType>
                                                      </xs:element>
                                                      <xs:element name="EntityId">
                                                        <xs:complexType>
                                                          <xs:sequence>
```

```
                                                        <xs:element name="Id"
    type="xs:string" />
                                                        <xs:element name="Name"
    type="xs:string" />
                                                        <xs:element name="InternalId"
    type="xs:unsignedByte" />
                                                      </xs:sequence>
                                                    </xs:complexType>
                                                  </xs:element>
                                                  <xs:element name="MemberType"
    type="xs:string" />
                                                    </xs:sequence>
                                                  </xs:complexType>
                                                </xs:element>
                                                <xs:element name="Permission"
    type="xs:string" />
                                                <xs:element name="Attributes">
                                                  <xs:complexType>
                                                    <xs:sequence>
                                                      <xs:element maxOccurs="unbounded"
    name="MetadataAttribute">
                                                        <xs:complexType>
                                                          <xs:sequence>
                                                            <xs:element name="Identifier">
                                                              <xs:complexType>
                                                                <xs:sequence>
                                                                  <xs:element name="Id"
    type="xs:string" />
                                                                  <xs:element name="Name"
    type="xs:string" />
                                                                  <xs:element
    name="InternalId" type="xs:unsignedShort" />
                                                                  <xs:element
    name="ModelId">
                                                                    <xs:complexType>
                                                                      <xs:sequence>
                                                                        <xs:element
    name="Id" type="xs:string" />
                                                                        <xs:element
    name="Name" type="xs:string" />
                                                                        <xs:element
    name="InternalId" type="xs:unsignedByte" />
                                                                      </xs:sequence>
                                                                    </xs:complexType>
                                                                  </xs:element>
                                                                  <xs:element
    name="EntityId">
                                                                    <xs:complexType>
                                                                      <xs:sequence>
                                                                        <xs:element
    name="Id" type="xs:string" />
                                                                        <xs:element
    name="Name" type="xs:string" />
                                                                        <xs:element
    name="InternalId" type="xs:unsignedByte" />
                                                                      </xs:sequence>
                                                                    </xs:complexType>
                                                                  </xs:element>
                                                                  <xs:element
    name="MemberType" type="xs:string" />
                                                                </xs:sequence>
                                                              </xs:complexType>
                                                            </xs:element>
                                                            <xs:element name="Permission"
    type="xs:string" />
                                                            <xs:element
    name="AttributeType" type="xs:string" />
                                                            <xs:element
    name="ChangeTrackingGroup" type="xs:unsignedByte" />
```

```
                                                                <xs:element name="DataType"
    type="xs:string" />
                                                                <xs:element
    name="DataTypeInformation" nillable="true" />
                                                                <xs:element name="DisplayWidth"
    type="xs:unsignedByte" />
                                                                <xs:element
    name="DomainEntityId">
                                                                  <xs:complexType>
                                                                    <xs:sequence>
                                                                      <xs:element name="Id"
    type="xs:string" />
                                                                      <xs:element name="Name"
    type="xs:string" />
                                                                      <xs:element
    name="InternalId" type="xs:unsignedByte" />
                                                                    </xs:sequence>
                                                                  </xs:complexType>
                                                                </xs:element>
                                                                <xs:element
    name="DomainEntityIsFlat" type="xs:boolean" />
                                                                <xs:element
    name="DomainEntityPermission" type="xs:string" />
                                                                <xs:element
    name="FullyQualifiedName" nillable="true" />
                                                                <xs:element name="InputMaskId">
                                                                  <xs:complexType>
                                                                    <xs:sequence>
                                                                      <xs:element name="Id"
    type="xs:string" />
                                                                      <xs:element name="Name"
    type="xs:string" />
                                                                      <xs:element
    name="InternalId" type="xs:unsignedByte" />
                                                                    </xs:sequence>
                                                                  </xs:complexType>
                                                                </xs:element>
                                                                <xs:element name="IsCode"
    type="xs:boolean" />
                                                                <xs:element name="IsName"
    type="xs:boolean" />
                                                                <xs:element name="IsReadOnly"
    type="xs:boolean" />
                                                                <xs:element name="IsSystem"
    type="xs:boolean" />
                                                                <xs:element name="SortOrder"
    type="xs:unsignedByte" />
                                                              </xs:sequence>
                                                            </xs:complexType>
                                                          </xs:element>
                                                        </xs:sequence>
                                                      </xs:complexType>
                                                    </xs:element>
                                                    <xs:element name="FullName"
    type="xs:string" />
                                                    <xs:element name="SortOrder"
    type="xs:unsignedByte" />
                                                  </xs:sequence>
                                                </xs:complexType>
                                              </xs:element>
                                            </xs:sequence>
                                          </xs:complexType>
                                        </xs:element>
                                        <xs:element name="Attributes">
                                          <xs:complexType>
                                            <xs:sequence>
                                              <xs:element maxOccurs="unbounded"
    name="MetadataAttribute">
                                                <xs:complexType>
```

```
                                                    <xs:sequence>
                                                      <xs:element name="Identifier">
                                                        <xs:complexType>
                                                          <xs:sequence>
                                                            <xs:element name="Id"
    type="xs:string" />
                                                            <xs:element name="Name"
    type="xs:string" />
                                                            <xs:element name="InternalId"
    type="xs:unsignedShort" />
                                                            <xs:element name="ModelId">
                                                              <xs:complexType>
                                                                <xs:sequence>
                                                                  <xs:element name="Id"
    type="xs:string" />
                                                                  <xs:element name="Name"
    type="xs:string" />
                                                                  <xs:element name="InternalId"
    type="xs:unsignedByte" />
                                                                </xs:sequence>
                                                              </xs:complexType>
                                                            </xs:element>
                                                            <xs:element name="EntityId">
                                                              <xs:complexType>
                                                                <xs:sequence>
                                                                  <xs:element name="Id"
    type="xs:string" />
                                                                  <xs:element name="Name"
    type="xs:string" />
                                                                  <xs:element name="InternalId"
    type="xs:unsignedByte" />
                                                                </xs:sequence>
                                                              </xs:complexType>
                                                            </xs:element>
                                                            <xs:element name="MemberType"
    type="xs:string" />
                                                          </xs:sequence>
                                                        </xs:complexType>
                                                      </xs:element>
                                                      <xs:element name="Permission"
    type="xs:string" />
                                                      <xs:element name="AttributeType"
    type="xs:string" />
                                                      <xs:element name="ChangeTrackingGroup"
    type="xs:unsignedByte" />
                                                      <xs:element name="DataType"
    type="xs:string" />
                                                      <xs:element name="DataTypeInformation"
    type="xs:unsignedByte" />
                                                      <xs:element name="DisplayWidth"
    type="xs:unsignedByte" />
                                                      <xs:element name="DomainEntityId">
                                                        <xs:complexType>
                                                          <xs:sequence>
                                                            <xs:element name="Id"
    type="xs:string" />
                                                            <xs:element name="Name"
    type="xs:string" />
                                                            <xs:element name="InternalId"
    type="xs:unsignedByte" />
                                                          </xs:sequence>
                                                        </xs:complexType>
                                                      </xs:element>
                                                      <xs:element name="DomainEntityIsFlat"
    type="xs:boolean" />
                                                      <xs:element name="DomainEntityPermission"
    type="xs:string" />
                                                      <xs:element name="FullyQualifiedName"
    type="xs:string" />
```

```
                                                    <xs:element name="InputMaskId">
                                                      <xs:complexType>
                                                        <xs:sequence>
                                                          <xs:element name="Id"
    type="xs:string" />
                                                          <xs:element name="Name"
    type="xs:string" />
                                                          <xs:element name="InternalId"
    type="xs:unsignedByte" />
                                                        </xs:sequence>
                                                      </xs:complexType>
                                                    </xs:element>
                                                    <xs:element name="IsCode" type="xs:boolean"
    />
                                                    <xs:element name="IsName" type="xs:boolean"
    />
                                                    <xs:element name="IsReadOnly"
    type="xs:boolean" />
                                                    <xs:element name="IsSystem"
    type="xs:boolean" />
                                                    <xs:element name="SortOrder"
    type="xs:unsignedByte" />
                                                  </xs:sequence>
                                                </xs:complexType>
                                              </xs:element>
                                            </xs:sequence>
                                          </xs:complexType>
                                        </xs:element>
                                        <xs:element name="EntityId">
                                          <xs:complexType>
                                            <xs:sequence>
                                              <xs:element name="Id" type="xs:string" />
                                              <xs:element name="Name" type="xs:string" />
                                              <xs:element name="InternalId"
    type="xs:unsignedByte" />
                                            </xs:sequence>
                                          </xs:complexType>
                                        </xs:element>
                                        <xs:element name="Type" type="xs:string" />
                                      </xs:sequence>
                                    </xs:complexType>
                                  </xs:element>
                                </xs:sequence>
                              </xs:complexType>
                            </xs:element>
                          </xs:sequence>
                        </xs:complexType>
                      </xs:element>
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="ExplicitHierarchies">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element maxOccurs="unbounded" name="ExplicitHierarchy">
                        <xs:complexType>
                          <xs:sequence>
                            <xs:element name="Identifier">
                              <xs:complexType>
                                <xs:sequence>
                                  <xs:element name="Id" type="xs:string" />
                                  <xs:element name="Name" type="xs:string" />
                                  <xs:element name="InternalId" type="xs:unsignedByte" />
                                  <xs:element name="ModelId">
                                    <xs:complexType>
                                      <xs:sequence>
                                        <xs:element name="Id" type="xs:string" />
                                        <xs:element name="Name" type="xs:string" />
                                        <xs:element name="InternalId" type="xs:unsignedByte" />
```

```
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
                <xs:element name="EntityId">
                  <xs:complexType>
                    <xs:sequence>
                      <xs:element name="Id" type="xs:string" />
                      <xs:element name="Name" type="xs:string" />
                      <xs:element name="InternalId" type="xs:unsignedByte" />
                    </xs:sequence>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
          <xs:element name="Permission" type="xs:string" />
          <xs:element name="FullyQualifiedName" type="xs:string" />
          <xs:element name="IsMandatory" type="xs:boolean" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
 </xs:complexType>
</xs:element>
<xs:element name="IsAdministrator" type="xs:boolean" />
<xs:element name="IsSystem" type="xs:boolean" />
<xs:element name="VersionFlags">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="VersionFlag">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="Identifier">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Id" type="xs:string" />
                  <xs:element name="Name" type="xs:string" />
                  <xs:element name="InternalId" type="xs:unsignedByte" />
                  <xs:element name="ModelId">
                    <xs:complexType>
                      <xs:sequence>
                        <xs:element name="Id" type="xs:string" />
                        <xs:element name="Name" type="xs:string" />
                        <xs:element name="InternalId" type="xs:unsignedByte" />
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Permission" type="xs:string" />
            <xs:element name="AssignedVersionId">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="Id" type="xs:string" />
                  <xs:element name="Name" />
                  <xs:element name="InternalId" type="xs:unsignedByte" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
            <xs:element name="Description" type="xs:string" />
            <xs:element name="IsCommittedOnly" type="xs:boolean" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Versions">
```

```
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="Version">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="Identifier">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Id" type="xs:string" />
                                <xs:element name="Name" type="xs:string" />
                                <xs:element name="InternalId" type="xs:unsignedByte" />
                                <xs:element name="ModelId">
                                  <xs:complexType>
                                    <xs:sequence>
                                      <xs:element name="Id" type="xs:string" />
                                      <xs:element name="Name" type="xs:string" />
                                      <xs:element name="InternalId" type="xs:unsignedByte" />
                                    </xs:sequence>
                                  </xs:complexType>
                                </xs:element>
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="Permission" type="xs:string" />
                          <xs:element name="CopiedFromVersionId">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Id" type="xs:string" />
                                <xs:element name="Name" />
                                <xs:element name="InternalId" type="xs:unsignedByte" />
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="Description" type="xs:string" />
                          <xs:element name="ValidationStatus" type="xs:string" />
                          <xs:element name="VersionFlagId">
                            <xs:complexType>
                              <xs:sequence>
                                <xs:element name="Id" type="xs:string" />
                                <xs:element name="Name" />
                                <xs:element name="InternalId" type="xs:unsignedByte" />
                              </xs:sequence>
                            </xs:complexType>
                          </xs:element>
                          <xs:element name="VersionNumber" type="xs:unsignedByte" />
                          <xs:element name="VersionStatus" type="xs:string" />
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="VersionFlags" />
  <xs:element name="Versions" />
</xs:schema>
```

3.2 Model Deployment XSD for SQL Server 2016

```
<?xml version="1.0" encoding="utf-16"?>
<xs:schema
xmlns:tns="http://schemas.datacontract.org/2004/07/Microsoft.MasterDataServices.Deployment"
```

```
elementFormDefault="qualified"
targetNamespace="http://schemas.datacontract.org/2004/07/Microsoft.MasterDataServices.Deploym
ent" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09" />
  <xs:complexType name="Package">
    <xs:sequence>
      <xs:element name="businessRuleSet" nillable="true"
xmlns:q1="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
type="q1:BusinessRules" />
      <xs:element name="exportViews" nillable="true"
xmlns:q2="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
type="q2:ArrayOfExportView" />
      <xs:element name="masterData" nillable="true"
xmlns:q3="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
type="q3:ArrayOfEntityMembers" />
      <xs:element name="metadata" nillable="true"
xmlns:q4="http://schemas.microsoft.com/sqlserver/masterdataservices/2009/09"
type="q4:Metadata" />
      <xs:element name="settings" nillable="true" type="tns:PackageSettings" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="Package" nillable="true" type="tns:Package" />
  <xs:complexType name="PackageSettings">
    <xs:sequence>
      <xs:element name="assemblyVersion" nillable="true" type="xs:string" />
      <xs:element name="containsData" type="xs:boolean" />
      <xs:element name="createdBy" nillable="true" type="xs:string" />
      <xs:element name="createdDate" type="xs:dateTime" />
      <xs:element name="deploymentType" type="tns:DeploymentType" />
      <xs:element name="productVersion" nillable="true" type="xs:string" />
      <xs:element name="version" type="xs:int" />
    </xs:sequence>
  </xs:complexType>
  <xs:element name="PackageSettings" nillable="true" type="tns:PackageSettings" />
  <xs:simpleType name="DeploymentType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Complete" />
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="DeploymentType" nillable="true" type="tns:DeploymentType" />
</xs:schema>
```

4 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- The removal of a document from the documentation set.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the technical content of the document is identical to the last released version.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.
- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3 Appendix A: Model Deployment XSD	Added table and rearranged content to accommodate addition of XSD for SQL Server 2016.	Y	Content update.
3.2 Model Deployment XSD for SQL Server 2016	Added section.	Y	New content added.

5 Index