

[MS-BDCSP]: Business Data Catalog Database Protocol Specification

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's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.msp>). 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.
- **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
04/04/2008	0.1		Initial Availability
06/27/2008	1.0	Major	Revised and edited the technical content
12/12/2008	1.01	Editorial	Revised and edited the technical content
03/18/2009	1.02	Editorial	Revised and edited the technical content
07/13/2009	1.03	Major	Changes made for template compliance
08/28/2009	1.04	Editorial	Revised and edited the technical content
11/06/2009	1.05	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction	8
1.1 Glossary	8
1.2 References	9
1.2.1 Normative References	9
1.2.2 Informative References	9
1.3 Protocol Overview (Synopsis)	10
1.4 Relationship to Other Protocols	10
1.5 Prerequisites/Preconditions	10
1.6 Applicability Statement	11
1.7 Versioning and Capability Negotiation	11
1.8 Vendor-Extensible Fields	11
1.9 Standards Assignments	11
2 Messages	12
2.1 Transport	12
2.2 Common Data Types	12
2.2.1 Simple Data Types and Enumerations	12
2.2.2 Common Fields	12
2.2.2.1 Id	12
2.2.2.2 Name	12
2.2.2.3 IsCached	12
2.2.2.4 EstimatedInstanceCount	12
2.2.2.5 MetadataObjectType	12
2.2.2.6 Position	14
2.2.2.7 IsDisplayed	14
2.2.2.8 IsCollection	14
2.2.2.9 IsOpenedInNewWindow	14
2.2.2.10 Icon	14
2.2.2.11 URL	14
2.2.2.12 Index	14
2.2.2.13 FilterDescriptorTypeName	14
2.2.2.14 IdentifierTypeName	15
2.2.2.15 MethodInstanceType	16
2.2.2.16 Direction	17
2.2.2.17 TypeReflectorTypeName	17
2.2.2.18 TypeDescriptorTypeName	17
2.2.2.19 ConnectionManagerTypeName	17
2.2.2.20 SystemUtilityTypeName	18
2.2.2.21 EntityInstanceTypeName	18
2.2.2.22 MetadataRights	18
2.2.2.23 IsStatic	19
2.2.3 Bit Fields and Flag Structures	19
2.2.4 Binary Structures	19
2.2.5 Result Sets	19
2.2.5.1 Action Result Set	19
2.2.5.2 ActionParameter Result Set	20
2.2.5.3 Association Result Set	20
2.2.5.4 Count Result Set	21
2.2.5.5 DataClass Result Set	21
2.2.5.6 Entity Result Set	22

2.2.5.7	FilterDescriptor Result Set	22
2.2.5.8	Identifier Result Set	23
2.2.5.9	Method Result Set	23
2.2.5.10	MethodInstance Result Set.....	24
2.2.5.11	Parameter Result Set	24
2.2.5.12	System Instance Result Set.....	25
2.2.5.13	System Result Set	25
2.2.5.14	TypeDescriptor Result Set.....	26
2.2.6	Tables and Views	27
2.2.7	XML Structures	27
3	Protocol Details.....	28
3.1	Back End Database Server Details	28
3.1.1	Abstract Data Model	28
3.1.2	Timers	29
3.1.3	Initialization	29
3.1.4	Higher-Layer Triggered Events.....	29
3.1.5	Message Processing Events and Sequencing Rules.....	29
3.1.5.1	proc_ar_AddOrInsertLocalizedNameForMetadataObjectId	29
3.1.5.2	proc_ar_AddOrInsertPropertyForMetadataObjectId	30
3.1.5.3	proc_ar_BumpCacheInvalidationCounter	31
3.1.5.4	proc_ar_ClearAccessControlEntriesForMetadataObject.....	32
3.1.5.5	proc_ar_CopyAccessControlEntriesForMetadataObjectId	32
3.1.5.6	proc_ar_CreateAction.....	32
3.1.5.7	proc_ar_CreateActionParameter	34
3.1.5.8	proc_ar_CreateAssociation.....	34
3.1.5.9	proc_ar_CreateEntity	36
3.1.5.10	proc_ar_CreateFilterDescriptor	37
3.1.5.11	proc_ar_CreateIdentifier	37
3.1.5.12	proc_ar_CreateMethod	38
3.1.5.13	proc_ar_CreateMethodInstance	39
3.1.5.14	proc_ar_CreateParameter	41
3.1.5.15	proc_ar_CreateSystem	42
3.1.5.16	proc_ar_CreateSystemInstance	43
3.1.5.17	proc_ar_CreateTypeDescriptor	43
3.1.5.18	proc_ar_DeleteActionById.....	45
3.1.5.19	proc_ar_DeleteActionParameterById	46
3.1.5.20	proc_ar_DeleteAssociationById.....	46
3.1.5.21	proc_ar_DeleteDefaultValue	47
3.1.5.22	proc_ar_DeleteEntityById	48
3.1.5.23	proc_ar_DeleteFilterDescriptorById	49
3.1.5.24	proc_ar_DeleteIdentifierById.....	49
3.1.5.25	proc_ar_DeleteLocalizedNameForMetadataObjectByLCID	50
3.1.5.26	proc_ar_DeleteLocalizedNamesByMetadataObjectId	51
3.1.5.27	proc_ar_DeleteMethodById	51
3.1.5.28	proc_ar_DeleteMethodInstanceById	52
3.1.5.29	proc_ar_DeleteParameterById	53
3.1.5.30	proc_ar_DeletePropertiesById	54
3.1.5.31	proc_ar_DeletePropertyForMetadataObjectId.....	54
3.1.5.32	proc_ar_DeleteSystemById	55
3.1.5.33	proc_ar_DeleteSystemInstanceById	56
3.1.5.34	proc_ar_DeleteTypeDescriptorById	56
3.1.5.35	proc_ar_EnsureApplicationRegistryExists	57

3.1.5.36	proc_ar_GetAccessControlEntriesForMetadataObject	57
3.1.5.36.1	Access Control Entry Result Set	58
3.1.5.37	proc_ar_GetActionById	58
3.1.5.37.1	Action Result Set	59
3.1.5.38	proc_ar_GetActionParameterById	59
3.1.5.38.1	ActionParameter Result Set	59
3.1.5.39	proc_ar_GetActionParametersForActionWithCount	59
3.1.5.39.1	Count Result Set	59
3.1.5.39.2	ActionParameter Result Set	59
3.1.5.40	proc_ar_GetActionsForEntityWithCount	60
3.1.5.40.1	Count Result Set	60
3.1.5.40.2	Action Result Set	60
3.1.5.41	proc_ar_GetAllLocalizedNamesForMetadataObjectWithCount	60
3.1.5.41.1	Count Result Set	60
3.1.5.41.2	Localized Name Result Set	60
3.1.5.42	proc_ar_GetAllSystemInstancesLikeNameWithCount	61
3.1.5.42.1	Count Result Set	61
3.1.5.42.2	System Instance Result Set	62
3.1.5.43	proc_ar_GetAllSystemInstancesWithCount	62
3.1.5.43.1	Count Result Set	62
3.1.5.43.2	System Instance Result Set	62
3.1.5.44	proc_ar_GetAllSystemsWithCount	62
3.1.5.44.1	Count Result Set	62
3.1.5.44.2	System Result Set	62
3.1.5.45	proc_ar_GetAssociationById	62
3.1.5.45.1	Association Result Set	63
3.1.5.46	proc_ar_GetAssociationByName	63
3.1.5.46.1	Association Result Set	63
3.1.5.47	proc_ar_GetAssociationsForDataClassWithCount	63
3.1.5.47.1	Count Result Set	64
3.1.5.47.2	Association Result Set	64
3.1.5.48	proc_ar_GetAssociationsForEntityAndRoleWithCount	64
3.1.5.48.1	Count Result Set	64
3.1.5.48.2	Association Result Set	64
3.1.5.49	proc_ar_GetAssociationsForMethodWithCount	65
3.1.5.49.1	Count Result Set	65
3.1.5.49.2	Association Result Set	65
3.1.5.50	proc_ar_GetCacheInvalidationCountersWithCount	65
3.1.5.50.1	Count Result Set	65
3.1.5.50.2	Cache Version Stamps Result Set	65
3.1.5.51	proc_ar_GetChildTypeDescriptorsForTypeDescriptorWithCount	66
3.1.5.51.1	Count Result Set	66
3.1.5.51.2	TypeDescriptor Result Set	66
3.1.5.52	proc_ar_GetDataClassById	66
3.1.5.52.1	DataClass Result Set	67
3.1.5.53	proc_ar_GetDataClassesForSystemWithCount	67
3.1.5.53.1	Count Result Set	67
3.1.5.53.2	DataClass Result Set	67
3.1.5.54	proc_ar_GetDefaultValuesForTypeDescriptor	67
3.1.5.54.1	DefaultValues Result Set	68
3.1.5.55	proc_ar_GetDependentEntitiesForEntity	68
3.1.5.55.1	EntityId Result Set	69
3.1.5.56	proc_ar_GetEntitiesForAssociationAndRoleWithCount	69

3.1.5.56.1	Count Result Set	69
3.1.5.56.2	Entity Result Set	69
3.1.5.57	proc_ar_GetEntitiesForSystemLikeNameWithCount	69
3.1.5.57.1	Count Result Set	70
3.1.5.57.2	Entity Result Set	70
3.1.5.58	proc_ar_GetEntitiesForSystemWithCount	70
3.1.5.58.1	Count Result Set	70
3.1.5.58.2	Entity Result Set	71
3.1.5.59	proc_ar_GetEntityById	71
3.1.5.59.1	Entity Result Set	71
3.1.5.60	proc_ar_GetFilterDescriptorById	71
3.1.5.60.1	FilterDescriptor Result Set	71
3.1.5.61	proc_ar_GetFilterDescriptorsForMethodWithCount	71
3.1.5.61.1	Count Result Set	72
3.1.5.61.2	FilterDescriptor Result Set	72
3.1.5.62	proc_ar_GetIdentifierById	72
3.1.5.62.1	Identifier Result Set	72
3.1.5.63	proc_ar_GetIdentifiersForEntityWithCount	72
3.1.5.63.1	Count Result Set	73
3.1.5.63.2	Identifier Result Set	73
3.1.5.64	proc_ar_GetMethodById	73
3.1.5.64.1	Method Result Set	73
3.1.5.65	proc_ar_GetMethodInstanceById	73
3.1.5.65.1	MethodInstance Result Set	74
3.1.5.66	proc_ar_GetMethodInstancesForDataClassWithCount	74
3.1.5.66.1	Count Result Set	74
3.1.5.66.2	MethodInstance Result Set	74
3.1.5.67	proc_ar_GetMethodInstancesForMethodWithCount	74
3.1.5.67.1	Count Result Set	75
3.1.5.67.2	MethodInstance Result Set	75
3.1.5.68	proc_ar_GetMethodsForDataClassWithCount	75
3.1.5.68.1	Count Result Set	75
3.1.5.68.2	Method Result Set	75
3.1.5.69	proc_ar_GetParameterById	75
3.1.5.69.1	Parameter Result Set	76
3.1.5.70	proc_ar_GetParametersForMethodWithCount	76
3.1.5.70.1	Count Result Set	76
3.1.5.70.2	Parameter Result Set	76
3.1.5.71	proc_ar_GetPropertiesForMetadataObject	76
3.1.5.71.1	Property Result Set	77
3.1.5.72	proc_ar_GetRootTypeDescriptorForParameter	77
3.1.5.72.1	TypeDescriptor Result Set	77
3.1.5.73	proc_ar_GetSystemById	77
3.1.5.73.1	System Result Set	78
3.1.5.74	proc_ar_GetSystemDataBySystemName	78
3.1.5.74.1	System Data Result Set	78
3.1.5.75	proc_ar_GetSystemInstanceById	78
3.1.5.75.1	System Instance Result Set	79
3.1.5.76	proc_ar_GetSystemInstancesForSystemWithCount	79
3.1.5.76.1	Count Result Set	79
3.1.5.76.2	System Instance Result Set	79
3.1.5.77	proc_ar_GetSystemsLikeNameWithCount	79
3.1.5.77.1	Count Result Set	80

3.1.5.77.2	System Result Set	80
3.1.5.78	proc_ar_GetTypeDescriptorById	80
3.1.5.78.1	TypeDescriptor Result Set	80
3.1.5.79	proc_ar_GetTypeDescriptorsByNameAndParameter	80
3.1.5.79.1	TypeDescriptor Result Set	81
3.1.5.80	proc_ar_GetTypeDescriptorsForFilterDescriptorWithCount	81
3.1.5.80.1	Count Result Set	81
3.1.5.80.2	TypeDescriptor Result Set	81
3.1.5.81	proc_ar_SetAccessControlEntryForMetadataObject	81
3.1.5.82	proc_ar_SetDefaultAction	82
3.1.5.83	proc_ar_SetDefaultValuesForTypeDescriptor	83
3.1.5.84	proc_ar_SetSystemDataBySystemName	84
3.1.5.85	proc_ar_UpdateActionById	84
3.1.5.86	proc_ar_UpdateActionParameterById	85
3.1.5.87	proc_ar_UpdateAssociationById	86
3.1.5.88	proc_ar_UpdateEntityById	87
3.1.5.89	proc_ar_UpdateFilterDescriptorById	88
3.1.5.90	proc_ar_UpdateIdentifierById	89
3.1.5.91	proc_ar_UpdateMethodById	90
3.1.5.92	proc_ar_UpdateMethodInstanceById	91
3.1.5.93	proc_ar_UpdateParameterById	93
3.1.5.94	proc_ar_UpdateSystemById	94
3.1.5.95	proc_ar_UpdateSystemInstanceById	95
3.1.5.96	proc_ar_UpdateTypeDescriptorById	96
3.1.6	Timer Events	98
3.1.7	Other Local Events	98
3.2	Metadata Client Details	98
3.2.1	Abstract Data Model	98
3.2.1.1	MetadataObject Caching	98
3.2.2	Timers	99
3.2.3	Initialization	99
3.2.4	Higher-Layer Triggered Events	99
3.2.5	Message Processing Events and Sequencing Rules	99
3.2.6	Timer Events	99
3.2.7	Other Local Events	99
4	Protocol Examples	100
4.1	Creating an Entity	100
4.2	Reading the Security Information of a MetadataObject	101
4.3	Reading an Entity	101
4.4	Updating an Entity	102
4.5	Deleting an Entity	102
4.6	Cache Invalidation	103
5	Security	105
5.1	Security Considerations for Implementers	105
5.2	Index of Security Parameters	105
6	Appendix A: Product Behavior	106
7	Change Tracking	111
8	Index	113

1 Introduction

This document specifies the Business Data Catalog Database Protocol. This protocol provides an interface for protocol clients to store and retrieve information about interfaces of line-of-business systems (LOB systems) and annotations of these interfaces.

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

- access control entry (ACE)**
- globally unique identifier (GUID)**
- language code identifier (LCID)**
- security identifier (SID)**

The following terms are defined in [\[MS-OFCGLOS\]](#):

- AccessChecker**
- Action**
- ActionParameter**
- Association**
- back-end database server**
- business logic**
- Business Logic Module**
- ComparisonFilter**
- DataClass**
- Entity**
- FilterDescriptor**
- Finder**
- front-end Web server**
- GenericInvoker**
- Identifier**
- IdEnumerator**
- LastIdFilter**
- LimitFilter**
- line-of-business (LOB) system**
- LobSystem**
- LobSystemInstance**
- locale**
- metadata model**
- metadata store**
- MetadataObject**
- MetadataObjectId**
- Method**
- MethodInstance**
- Parameter**
- PasswordCredentialFilter**
- Property**
- RangeFilter**
- result set**
- return code**
- ReturnTypeDescriptor**
- root TypeDescriptor**
- Scalar**
- security principal**

SpecificFinder
SsoTicketFilter
stored procedure
T-SQL (Transact-Structured Query Language)
TypeDescriptor
TypeReflector
URL (Uniform Resource Locator)
UserContextFilter
UsernameCredentialFilter
UserProfileFilter
ViewAccessor
WildcardFilter

The following terms are specific to this document:

Business Logic Module Reference: A string of characters that identifies a specific Business Logic Module.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[ECMA-335] ECMA international, "Common Language Infrastructure (CLI) Partitions I to VI", ECMA-335, June 2006, <http://www.ecma-international.org/publications/standards/Ecma-335.htm>

[MSDN-TSQL-Ref] Microsoft Corporation, "Transact-SQL Reference", [http://msdn.microsoft.com/en-us/library/ms189826\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms189826(SQL.90).aspx)

[MS-SQL] Microsoft Corporation, "SQL Server 2000 Architecture and XML/Internet Support", Volume 1 of Microsoft SQL Server 2000 Reference Library, Microsoft Press, 2001, ISBN 0-7356-1280-3, [http://msdn.microsoft.com/en-us/library/dd631854\(v=SQL.10\).aspx](http://msdn.microsoft.com/en-us/library/dd631854(v=SQL.10).aspx)

[MS-TDS] Microsoft Corporation, "[Tabular Data Stream Protocol Specification](#)", February 2008.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>

1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)", June 2008.

1.3 Protocol Overview (Synopsis)

Enterprises have a variety of data stored in various **line-of-business (LOB) system**. Typically, this data is accessible only through the proprietary programming interface of these software systems. It is desirable to be able to provide access to such data via a set of normalized interfaces so that users do not have to learn system-specific or adapter-specific programming patterns for each software system.

To facilitate this, it is possible to store descriptions of the programmatic interface of the LOB systems using data structures such as **Methods, Parameters, and TypeDescriptors**, along with information about the LOB systems themselves (such as the server name, connection string and how to authenticate), using data structures such as **LobSystem** and **LobSystemInstance**. Methods can be considered to live within an **Entity** abstraction, representing a business data type, such as Customer or Order. The LOB system interface definitions can then be transformed into normalized, stereotypical operations against Entities such as 'Read-An-Entity-Instance-By-Id', 'Read-Entity-Instances' and 'Check-Entity-Instance-Permissions' by annotating the actual LOB system interface descriptions, with the annotations described by data structures such as **MethodInstance, Identifier, FilterDescriptor, and Association**. These data structures, collectively called **MetadataObjects**, can be grouped into related collections called **metadata models** that describe a single LOB system. Once a store of metadata models is made available, it can be utilized by a runtime engine to convert a stereotypical, normalized operation requested by an application that uses the protocol client into a LOB system-specific invocation programmatically.

This protocol allows a metadata client to create, read, update and delete **MetadataObjects** in a **metadata store**. For write operations, the protocol server will provide validation and diagnostic errors such that protocol clients can maintain the set of stored **MetadataObjects** in a state that satisfies certain semantic constraints for metadata models. These constraints are documented in detail in Protocol Details, section [3](#).

1.4 Relationship to Other Protocols

The following diagram shows the transport stack that the protocol uses:

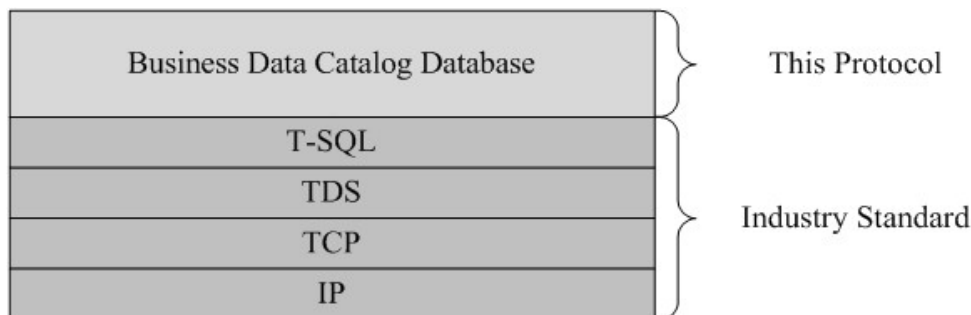


Figure 1: This protocol in relation to other protocols

1.5 Prerequisites/Preconditions

The operations described by the protocol operate between a client and a **back-end database server** on that the databases are stored. The client is expected to know the location and connection information for the databases.

This protocol requires that the protocol client has appropriate permissions to call the **stored procedures** stored on the back-end database server.

1.6 Applicability Statement

There are typically two types of applications that can be built using the **Metadata Client**, though an application that combines these functions in a single implementation is also feasible:

Metadata model designers, whose primary purpose is to create or edit a **Metadata model**. These applications typically offer some graphical design surface and connectivity to LOB systems of known types to enable mining of the LOB system public interface definition and creation of corresponding **MetadataObjects** in the protocol server store.

Metadata model consumers, whose primary purpose is to read a **Metadata model** in the protocol server store and use the information therein to convert uniform, stereotypical operations into LOB system-specific interface invocations.

This protocol does not specify how the stored **MetadataObjects** can be used to do the conversion from a stereotypical client request into a system-specific invocation; it is merely a **MetadataObject** storage and retrieval protocol.

This protocol is intended for use by protocol clients and protocol servers that are both connected by high-bandwidth, low latency network connections.

1.7 Versioning and Capability Negotiation

- **Security and Authentication Methods:** This protocol supports the Security Support Provider Interface (SSPI) and SQL Authentication with the Protocol Server role specified in [\[MS-TDS\]](#).

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

[MS-TDS] is the transport protocol used to call the stored procedures, **return codes**, and return **result sets**.

2.2 Common Data Types

The following sections define the common data types that are used in this protocol.

2.2.1 Simple Data Types and Enumerations

None.

2.2.2 Common Fields

2.2.2.1 Id

Id: int NOT NULL. Identifies a **MetadataObject** uniquely within a metadata store.

2.2.2.2 Name

Name: nvarchar(255) NOT NULL. The programmatic name of a **MetadataObject**.

2.2.2.3 IsCached

IsCached: bit NOT NULL. A bit that specifies the frequency of the use of a MetadataObject. Protocol clients can use this as a recommendation as to whether to cache a **MetadataObject** in memory.

Value	Description
0	The MetadataObject is infrequently used.
1	The MetadataObject is frequently used.

2.2.2.4 EstimatedInstanceCount

EstimatedInstanceCount: int NOT NULL. The maximum number of instances of an entity estimated to be returned from a physical LOB system represented by a **LobSystemInstance**.

2.2.2.5 MetadataObjectType

MetadataObjectType: nvarchar(255). The type of a MetadataObject. The value MUST be in the following table. If the length is greater than 250, then the characters after 250th character are ignored by the protocol server.

Value	Description
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.LobSystem	Identifies the LobSystem MetadataObject type.

Value	Description
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.LobSystemInstance	Identifies the LobSystemInstance MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Entity	Identifies the Entity MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Identifier	Identifies the Identifier MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Method	Identifies the Method MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.MethodInstance	Identifies the MethodInstance MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.FilterDescriptor	Identifies the FilterDescriptor MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Parameter	Identifies the Parameter MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.TypeDescriptor	Identifies the TypeDescriptor MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Action	Identifies the Action MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.ActionParameter	Identifies the ActionParameter MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.AccessControlledMetadataObject	Identifies the set of MetadataObject types that have access control entry (ACE) associated with them.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Association	Identifies the

Value	Description
	Association MetadataObject type.
Microsoft.Office.Server.ApplicationRegistry.MetadataModel.MetadataObject	Identifies the set of MetadataObject types that have Property and localized names associated with them.

2.2.2.6 Position

Position: tinyint NOT NULL. The order of an Action among the other Actions displayed in a user interface for an Entity.

2.2.2.7 IsDisplayed

IsDisplayed: bit NOT NULL. A bit that specifies whether an Action is displayed in the user interface presented to the user. The application that uses the protocol client typically uses this value as a guidance to represent the Action in the user interface.

2.2.2.8 IsCollection

IsCollection: bit NOT NULL. A bit that specifies whether a TypeDescriptor MUST be interpreted as a collection of native LOB System data structures.

2.2.2.9 IsOpenedInNewWindow

IsOpenedInNewWindow: bit NOT NULL. A bit that specifies whether the results of executing an Action are displayed in a new window in the user interface presented to the user. The application that uses the protocol client typically uses this value as guidance on creating new window when the Action is executed.

2.2.2.10 Icon

Icon: nvarchar(2080). The **URL** of the icon associated with an Action.

2.2.2.11 URL

URL: nvarchar(2080) NOT NULL. The URL associated with an Action.

2.2.2.12 Index

Index: tinyint NOT NULL. A value, indicating the position of an ActionParameter among the ActionParameters of the Action that contains the ActionParameter.

2.2.2.13 FilterDescriptorTypeName

FilterDescriptorTypeName: nvarchar(255) NOT NULL. The type of a FilterDescriptor. The value MUST be in the following table.

Value	Description
Microsoft.Office.Server.ApplicationRegistry.Runtime.LimitFilter	Indicates that a FilterDescriptor describes a LimitFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.EqualsFilter	Indicates that a FilterDescriptor describes a ComparisonFilter with its comparator set to '=='.
Microsoft.Office.Server.ApplicationRegistry.Runtime.WildcardFilter	Indicates that a FilterDescriptor describes a WildcardFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.RangeFilter	Indicates that a FilterDescriptor describes a RangeFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.UserContextFilter	Indicates that a FilterDescriptor describes a UserContextFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.UsernameCredentialFilter	Indicates that a FilterDescriptor describes a UsernameCredentialFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.PasswordCredentialFilter	Indicates that a FilterDescriptor describes a PasswordCredentialFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.LastIdFilter	Indicates that a FilterDescriptor describes a LastIdFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.SsoTicketFilter	Indicates that a FilterDescriptor describes a SsoTicketFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.UserProfileFilter	Indicates that a FilterDescriptor describes a UserProfileFilter .
Microsoft.Office.Server.ApplicationRegistry.Runtime.ComparisonFilter	Indicates that a FilterDescriptor describes a ComparisonFilter.

2.2.2.14 IdentifierTypeName

IdentifierTypeName: nvarchar(255) NOT NULL. The type of the identifiers of instances of an Entity returned from an LOB system. The value MUST be in the following table.

Value	Description
System.String	Defines the identifier of instances of the Entity to be strings of Unicode text.
System.Int16	Defines the identifier of instances of the Entity to be a number ranging from negative 32768 to positive 32767.

Value	Description
System.Int32	Defines the identifier of instances of the Entity to be a number ranging from negative 2,147,483,648 to positive 2,147,483,647.
System.Int64	Defines the identifier of instances of the Entity to be a number ranging from negative 9,223,372,036,854,775,808 to positive 9,223,372,036,854,775,807.
System.UInt16	Defines the identifier of instances of the Entity to be a number ranging from 0 to 65535.
System.UInt32	Defines the identifier of instances of the Entity to be a number ranging from 0 to 4,294,967,295.
System.UInt64	Defines the identifier of instances of the Entity to be a number ranging from 0 to 18,446,744,073,709,551,615.
System.DateTime	Defines the identifier of instances of the Entity to be a date and time ranging from 12:00:00 midnight, January 1, 1 Anno Domini (Common Era) to 11:59:59 P.M., December 31, 9999 Anno Domini (Common Era).
System.Single	Defines the identifier of instances of the Entity to be a single precision number ranging from negative 3.402823e38 to 3.402823e38.
System.Double	Defines the identifier of instances of the Entity to be a double precision number ranging from negative 1.79769313486232e308 to positive 1.79769313486232e308 as well as positive or negative zero, positive infinity, negative infinity and NaN.
System.Decimal	Defines the identifier of instances of the Entity to be a number ranging from negative 79,228,162,514,264,337,593,543,950,335 to positive 79,228,162,514,264,337,593,543,950,335.
System.Char	Defines the identifier of instances of the Entity to be a Unicode character.
System.Byte	Defines the identifier of instances of the Entity to be a number ranging from 0 to 255.
System.SByte	Defines the identifier of instances of the Entity to be a number ranging from negative 128 to positive 127.
Guid	Defines the identifier of instances of the Entity to be a GUID .

2.2.2.15 MethodInstanceType

MethodInstanceType: tinyint NOT NULL. The type of a MethodInstance. The value MUST be in the following table.

Value	Description
1	Indicates that a MethodInstance should be interpreted as a Finder by the protocol client.
2	Indicates that a MethodInstance should be interpreted as a SpecificFinder by the protocol client.
3	Indicates that a MethodInstance should be interpreted as a ViewAccessor by the protocol client.
4	Indicates that a MethodInstance should be interpreted as a GenericInvoker by the protocol client.
5	Indicates that a MethodInstance should be interpreted as an IdEnumerator by the protocol client.

Value	Description
6	Indicates that a MethodInstance should be interpreted as a Scalar by the protocol client.
7	Indicates that a MethodInstance should be interpreted as an AccessChecker by the protocol client.

2.2.2.16 Direction

Direction: tinyint NOT NULL. The direction of the Parameter while calling the Method that contains the Parameter. The value MUST be in the following table.

Value	Description
1	Used for input purposes only.
2	Used for output purposes only.
3	Used for input before calling the LOB system and then for reading the output data once the call is complete.
4	Used to indicate that a parameter is the formal return parameter.

2.2.2.17 TypeReflectorTypeName

TypeReflectorTypeName: nvarchar(255) NOT NULL. Identifies the **TypeReflector** associated with a Parameter. The value MUST be in the following table.

Value	Description
Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.WebService.WebServiceTypeReflector or	Used when modeling Web service-based LOB systems.
Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.Db.DbTypeReflector	Used when modeling database-based LOB systems.

2.2.2.18 TypeDescriptorTypeName

TypeDescriptorTypeName: nvarchar(255) NOT NULL. The name of a unit of implementation-specific [<1> business logic \(2\)](#) that exists in an implementation-specific [<2> Business Logic Module](#).

2.2.2.19 ConnectionManagerTypeName

ConnectionManagerTypeName: nvarchar(255) NOT NULL. The name of the connection manager that will be used while connecting to this LobSystem. The value MUST be in the following table.

Value	Description
Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.WebService.WebServiceConnectionManager	Used when modeling Web service-based LOB systems.
Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.Db.DbConnectionManager	Used when modeling database-based LOB systems.

2.2.2.20 SystemUtilityTypeName

SystemUtilityTypeName: nvarchar(255) NOT NULL. The name of the Implementation-Specific logic module that will be used to execute the Methods in this LobSystem. The value MUST be in the following table.

Value	Description
Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.WebService.WebServiceSystemUtility	Used when modeling Web service-based LOB systems.
Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.Db.DbSystemUtility	Used when modeling database-based LOB systems.

2.2.2.21 EntityInstanceTypeName

2.2.2.22 MetadataRights

MetadataRights: bigint NOT NULL. The permissions available to a **security principal (2)** for a MetadataObject. The value MUST be in the following table.

Value	Description
0x00	No permissions.
0x01	Ability to call a MethodInstance.
0x02	Ability to change the attributes of a MetadataObject or its relationship to other MetadataObjects.
0x04	Ability to change the permissions associated with a MetadataObject.
A positive	Implementation-specific abilities.

Value	Description
value	

2.2.2.23 IsStatic

IsStatic: bit NOT NULL. A bit that specifies whether the Method is associated with an EntityInstance. The value MUST be in the following table.

Value	Description
0	The Method operates in the context of a specific EntityInstance.
1	The Method operates out of the context of a specific EntityInstance.

2.2.3 Bit Fields and Flag Structures

None.

2.2.4 Binary Structures

None.

2.2.5 Result Sets

The following common result sets are used by this protocol.

2.2.5.1 Action Result Set

The Action Result Set contains information about Actions. Each row in the result set contains all the attributes of a single Action.

The **T-SQL** syntax for the result set is as follows:

```

Id                int,
EntityId          int,
Position         tinyint,
IsDisplayed       bit,
IsOpenedInNewWindow bit,
Icon              nvarchar(2080),
URL               nvarchar(2080),
Name              nvarchar(255),
IsCached          bit,
Version           int;

```

Id: The **MetadataObjectId** of the Action that is returned. The value MUST be an [Id](#).

EntityId: The MetadataObjectId of the Entity that contains this Action. The value MUST be an Id.

Position: The order of this Action among the other Actions displayed in a user interface for this Entity. The value MUST be a [Position](#).

IsDisplayed: A bit that provides a hint on whether this Action is displayed in the user interface presented to the user. The value MUST be an [IsDisplayed](#).

IsOpenedInNewWindow: A bit that provides a hint on whether the results of executing this Action are displayed in a new window in the user interface presented to the user. The value MUST be an [IsOpenInNewWindow](#).

Icon: The URL of the icon associated with the Action. The value MUST be an [Icon](#).

URL: The URL associated with the Action. The value MUST be a [URL](#).

Name: The name of the Action. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this Action is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this Action.

2.2.5.2 ActionParameter Result Set

The ActionParameter Result Set contains information about ActionParameters. Each row in the result set contains all the attributes of a single ActionParameter.

The T-SQL syntax for the result set is as follows:

```
Id                int,  
ActionId          int,  
Index             tinyint,  
Name              nvarchar(255),  
IsCached          bit,  
Version           int;
```

Id: The MetadataObjectId of the ActionParameter that is returned. The value MUST be an [Id](#).

ActionId: The MetadataObjectId of the Action that contains this ActionParameter. The value MUST be an [Id](#).

Index: A value indicating the position of this ActionParameter among the other ActionParameters in the Action that contains this ActionParameter. The value MUST be an [Index](#).

Name: The name of the ActionParameter. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this ActionParameter is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this ActionParameter.

2.2.5.3 Association Result Set

The Association Result Set contains information about Associations. Each row in the result set contains all the attributes of a single Association.

The T-SQL syntax for the result set is as follows:

```
Id                int,  
MethodId          int,  
ReturnTypeDescriptorId int,  
Type             tinyint,  
Name              nvarchar(255),
```

```
IsCached          bit,  
Version           int;
```

Id: The MetadataObjectId of the Association that is returned. The value MUST be an [Id](#).

MethodId: The MetadataObjectId of the Method that contains the Association. The value MUST be an Id.

ReturnTypeDescriptorId: The MetadataObjectId of the [ReturnTypeDescriptor](#). The value MUST be an Id.

Type: The type of the MethodInstance. The value MUST be a [MethodInstanceType](#).

Name: The name of the Association. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this Association is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this Association.

2.2.5.4 Count Result Set

The Count Result Set contains the number of rows expected in the subsequent result set. The stored procedures that return Count Result Set MUST also return a subsequent result set with rows that represent MetadataObjects, version stamp for the cache or localized names. The protocol client can use the Count Result Set to initialize collections with enough size to store items that will be returned in the immediately following result set.

The T-SQL syntax for the result set is as follows:

```
{rowCount}      int;
```

{rowCount}: The number of rows in the immediately following result set.

2.2.5.5 DataClass Result Set

The DataClass Result Set contains information about [DataClasses](#). Each row in the result set contains all the attributes of a single DataClass.

The T-SQL syntax for the result set is as follows:

```
Id              int,  
SystemId       int,  
Name           nvarchar(255),  
IsCached       bit,  
Version        int;
```

Id: The MetadataObjectId of the DataClass that is returned. The value MUST be an [Id](#).

SystemId: The MetadataObjectId of the LobSystem that contains this DataClass. The value MUST be an Id.

Name: The name of this DataClass. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this DataClass is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this DataClass.

2.2.5.6 Entity Result Set

The Entity Result Set contains information about Entities. Each row in the result set contains all the attributes of a single Entity.

The T-SQL syntax for the result set is as follows:

```
Id                int,  
EstimatedInstanceCount int,  
SystemId          int,  
Name              nvarchar(255),  
IsCached          bit,  
Version           int;
```

Id: The MetadataObjectId of the Entity that is returned. The value MUST be an [Id](#).

EstimatedInstanceCount: The maximum estimated number of instances of this Entity returned from the LobSystemInstance. The value MUST be an [EstimatedInstanceCount](#).

SystemId: The MetadataObjectId of the LobSystem that contains this Entity. The value MUST be an [Id](#).

Name: The name of this Entity. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this Entity is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this Entity.

2.2.5.7 FilterDescriptor Result Set

The FilterDescriptor Result Set contains information about FilterDescriptors. Each row in the result set contains all the attributes of a single FilterDescriptor.

The T-SQL syntax for the result set is as follows:

```
Id                int,  
TypeName          nvarchar(255),  
MethodId          int,  
Name              nvarchar(255),  
IsCached          bit,  
Version           int;
```

Id: The MetadataObjectId of the FilterDescriptor that is returned. The value MUST be an [Id](#).

TypeName: Identifies the type of a FilterDescriptor. The value MUST be a [FilterDescriptorTypeName](#).

MethodId: The MetadataObjectId of the Method that contains this FilterDescriptor. The value MUST be an [Id](#).

Name: The name of this FilterDescriptor. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this FilterDescriptor is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this FilterDescriptor.

2.2.5.8 Identifier Result Set

The Identifier Result Set contains information about Identifier. Each row in the result set contains all the attributes of a single Identifier.

The T-SQL syntax for the result set is as follows:

```
Id                int,
TypeName          nvarchar(255),
EntityId         int,
OrdinalNumber    tinyint,
Name             nvarchar(255),
IsCached         bit,
Version          int;
```

Id: The MetadataObjectId of the Identifier that is returned. The value MUST be an [Id](#).

TypeName: The type name of the identifiers of instances of an Entity returned from an LOB system. The value MUST be an [IdentifierTypeName](#).

EntityId: The MetadataObjectId of the Entity that contains this Identifier. The value MUST be an [Id](#).

OrdinalNumber: The sequence number for the Identifier that imposes a deterministic ordering of all Identifiers of an Entity. The rows in the result set MUST be ordered by increasing *OrdinalNumber*.

Name: The name of this Identifier. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this Identifier is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this Identifier.

2.2.5.9 Method Result Set

The Method Result Set contains information about Methods. Each row in the result set contains all the attributes of a single Method.

The T-SQL syntax for the result set is as follows:

```
Id                int,
ClassId          int,
IsStatic         bit,
Name            nvarchar(255),
IsCached         bit,
Version          int;
```

Id: The MetadataObjectId of the Method that is returned. The value MUST be an [Id](#).

ClassId: The MetadataObjectId of the DataClass of this Method.

IsStatic: A bit that specifies whether the Method is associated with an EntityInstance. The value MUST be an [IsStatic](#).

Name: The name of this Method. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this Method is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this Method.

2.2.5.10 MethodInstance Result Set

The MethodInstance Result Set contains information about MethodInstances. Each row in the result set contains all the attributes of a single MethodInstance.

The T-SQL syntax for the result set is as follows:

```
Id                int,  
MethodId          int,  
ReturnTypeDescriptorId int,  
Type              tinyint,  
Name              nvarchar(255),  
IsCached          bit,  
Version           int;
```

Id: The MetadataObjectId of the MethodInstance that is returned. The value MUST be an [Id](#).

MethodId: The MetadataObjectId of the Method that contains this MethodInstance. The value MUST be an [Id](#).

ReturnTypeDescriptorId: The MetadataObjectId of the ReturnTypeDescriptor. The value MUST be an [Id](#).

Type: The type of this MethodInstance. The value MUST be a [MethodInstanceType](#).

Name: The name of this MethodInstance. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this MethodInstance is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this MethodInstance.

2.2.5.11 Parameter Result Set

The Parameter Result Set contains information about Parameters. Each row in the result set contains all the attributes of a single Parameter.

The T-SQL syntax for the result set is as follows:

```
Id                int,  
MethodId          int,  
Direction         tinyint,  
OrdinalNumber     tinyint,  
TypeReflectorTypeName nvarchar(255),  
Name              nvarchar(255),  
IsCached          bit,
```



```
Version          int;
```

Id: The MetadataObjectId of the Parameter that is returned. The value MUST be an [Id](#).

MethodId: The MetadataObjectId of the Method that contains this Parameter. The value MUST be an Id.

Direction: The direction of the Parameter while calling the Method this Parameter is contained by. The value MUST be a [Direction](#).

OrdinalNumber: The sequence number that imposes a deterministic ordering of the Parameters while using them to call the Method this Parameter is contained by. The rows in the result set MUST be returned with OrdinalNumbers in ascending order.

TypeReflectorTypeName: The name of the TypeReflector associated with a Parameter. The value MUST be a [TypeReflectorTypeName](#).

Name: The name of this Parameter. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this Parameter is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this Parameter.

2.2.5.12 System Instance Result Set

The System Instance Result Set contains information about LobSystemInstances. Each row in the result set contains all the attributes of a single LobSystemInstance.

The T-SQL syntax for the result set is as follows:

```
Id              int,  
SystemId       int,  
Name           nvarchar(255),  
IsCached       bit,  
Version        int;
```

Id: The MetadataObjectId of the LobSystemInstance that is returned. The value MUST be an [Id](#).

SystemId: The MetadataObjectId of the LobSystem that contains this LobSystemInstance. The value MUST be an Id.

Name: The name of this LobSystemInstance. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this LobSystemInstance is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this LobSystemInstance.

2.2.5.13 System Result Set

The System Result Set contains information about LobSystems. Each row in the result set contains all the attributes of a single LobSystem.

The T-SQL syntax for the result set is as follows:

```

Id                int,
ConnectionFactoryTypeName  nvarchar(255),
SystemUtilityTypeName      nvarchar(255),
SystemEntityTypeNames     nvarchar(255),
Name                nvarchar(255),
IsCached           bit,
Version            int;

```

Id: The MetadataObjectId of the LobSystem that is returned. The value MUST be an [Id](#).

ConnectionFactoryTypeName: The name of the connection manager that will be used while connecting to this LOB system. The value MUST be a [ConnectionFactoryName](#).

SystemUtilityTypeName: The name of the System Utility that will be used to execute Methods in this LobSystem. The value MUST be a [SystemUtilityName](#).

SystemEntityTypeNames: The name of the unit of implementation-specific [<3>](#) business logic (2) that will be used to create the objects that will carry EntityInstance data to client applications. The value MUST be a [EntityTypeNames](#).

Name: The name of this LobSystem. The value MUST be a [Name](#).

IsCached: A bit that specifies whether this LobSystem is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this LobSystem.

2.2.5.14 TypeDescriptor Result Set

The TypeDescriptor Result Set contains information about TypeDescriptors. Each row in the result set contains all the attributes of a single TypeDescriptor.

The T-SQL syntax for the result set is as follows:

```

Id                int,
ParameterId       int,
ParentTypeDescriptorId  int,
TypeName          nvarchar(255),
InterpretedTypeName  nvarchar(255),
ContainsIdentifier bit,
IdentifierId       int,
ContainsFilterDescriptor  bit,
FilterDescriptorId  int,
IsCollection       bit,
Name              nvarchar(255),
IsCached           bit,
Version            int;

```

Id: The MetadataObjectId of the TypeDescriptor that is returned. The value MUST be an [Id](#).

ParameterId: The MetadataObjectId of the Parameter that contains this TypeDescriptor. The value MUST be an Id.

ParentTypeDescriptorId: The MetadataObjectId of the parent TypeDescriptor that contains this TypeDescriptor. The value MUST be an Id.

TypeName: The name of a unit of implementation-specific <4> business logic (2) that exists in an implementation-specific <5> business logic module. The value MUST be a [TypeDescriptorTypeName](#).

InterpretedTypeName: Same as the *TypeName*. The value MUST be a [TypeDescriptorTypeName](#).

ContainsIdentifier: A bit that specifies if any [TypeDescriptor](#) in the [TypeDescriptor](#) tree of this [TypeDescriptor](#) references an [Identifier](#).

IdentifierId: The [MetadataObjectId](#) of the [Identifier](#) associated with this [TypeDescriptor](#). If this is NULL, this [TypeDescriptor](#) does not have an associated [Identifier](#). If this is not NULL, *ContainsIdentifier* MUST be 1.

ContainsFilterDescriptor: A bit that specifies if any [TypeDescriptor](#) in the [TypeDescriptor](#) tree of this [TypeDescriptor](#) has an associated [FilterDescriptor](#).

FilterDescriptorId: The [MetadataObjectId](#) of the [FilterDescriptor](#) associated with this [TypeDescriptor](#). If this is NULL, this [TypeDescriptor](#) does not have an associated [FilterDescriptor](#). If this is not NULL, *ContainsFilterDescriptor* MUST be 1.

IsCollection: A bit that specifies whether this [TypeDescriptor](#) is to be interpreted by protocol clients as a collection of native LOB System data structures. The value MUST be an [IsCollection](#).

Name: The name of this [TypeDescriptor](#). The value MUST be a [Name](#).

IsCached: A bit that specifies whether this [TypeDescriptor](#) is frequently used. The value MUST be an [IsCached](#).

Version: The object version of this [TypeDescriptor](#).

2.2.6 Tables and Views

None.

2.2.7 XML Structures

No common XML Structures are defined in this protocol.

3 Protocol Details

3.1 Back End Database Server Details

The back-end database protocol server responds only to stored procedure calls from the protocol client. It returns result sets and return codes and never initiates communication with other endpoints of the protocol.

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

For this protocol, the back-end database server maintains lists to store the attributes of each of the following MetadataObjects:

- Entity
- DataClass
- LobSystem
- LobSystemInstance
- Method
- MethodInstance
- Parameter
- TypeDescriptor
- Association
- Identifier
- FilterDescriptor
- Action
- ActionParameter

The server MUST maintain a set of relations between these MetadataObject types. These relations are the following:

- Each LobSystemInstance is contained by a LobSystem.
- Each Entity is contained by a LobSystem.
- Each Method is contained by an Entity.
- Each MethodInstance is contained by a Method.
- Each Parameter is contained by a Method.

- Each TypeDescriptor that is not a **root TypeDescriptor** is contained by a TypeDescriptor.
- Each root TypeDescriptor is contained by a Parameter.
- Each FilterDescriptor is contained by a Method.
- Each Identifier is contained by an Entity.
- Each Association [<6>](#) is contained by a Method.
- Each DataClass is contained by a LobSystem.
- Each Action is contained by an Entity.
- Each ActionParameter is contained by an Action.

The implementations of the basic Create, Read, Update, and Delete stored procedures simply insert, read, update and delete items in each of these lists where the MetadataObjectId serves as the primary identifier.

The access control entries, localized names, and MetadataObject Properties are also stored in their own lists along with the MetadataObjectId of the associated MetadataObject.

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None

3.1.5 Message Processing Events and Sequencing Rules

The T-SQL syntax for each stored procedure and result set, and the variables they are composed of, is defined in the [\[MSDN-TSQL-Ref\]](#) protocol. In the T-SQL syntax, the variable name is followed by the type of the variable that can optionally have a length value in brackets and can optionally have a default value indicated by an equals sign followed by the default value. Unless otherwise specified, all stored procedures defined in this section are located in the metadata store.

For definitional clarity, a name has been assigned to any columns in the Result Sets that do not have a defined name in their current implementation. This does not affect the operation of the result set, as the ordinal position of any column with no defined name is expected by the **front-end Web server**. Such names are designated in the text using curly braces in the form `{name}`. The stored procedures that return multiple result sets should return them in the order they are specified. For example, `proc_ar_GetActionParametersForActionWithCount` should return the Count result set followed by the Action result set.

3.1.5.1 `proc_ar_AddOrInsertLocalizedNameForMetadataObjectId`

The `proc_ar_AddOrInsertLocalizedNameForMetadataObjectId` stored procedure is called to add a localized name for a MetadataObject in the specified locale. If a localized name already exists for the specified locale, it MUST be replaced by the specified localized name.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_AddOrInsertLocalizedNameForMetadataObjectId(  
    @MetadataObjectId      int,  
    @LocalizedName         nvarchar(255),  
    @LCID                  int,  
    @ErrorCode              int OUTPUT  
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject to which this localized name will be added or replaced. The value MUST be an [Id](#).

@LocalizedName: The localized name of this MetadataObject in the specified locale. If a localized name already exists for the specified locale and specified MetadataObject, it MUST be replaced by the specified localized name.

@LCID: The LCID representing the locale of the specified localized name.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-1	An error occurred while adding or replacing the specified localized name.
-3	The MetadataObject already contains the implementation-specific maximum allowed number of localized names.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.2 proc_ar_AddOrInsertPropertyForMetadataObjectId

The **proc_ar_AddOrInsertPropertyForMetadataObjectId** stored procedure is called to add a Property for a MetadataObject. If a Property with the specified name already exists for the specified MetadataObject, its value MUST be replaced by the specified value.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_AddOrInsertPropertyForMetadataObjectId(  
    @MetadataObjectId      int,  
    @Name                  nvarchar(255),  
    @Value                  sql_variant,  
    @ErrorCode              int OUTPUT  
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject to which this Property will be added or replaced. The value MUST be an [Id](#).

@Name: The programmatic name of the Property. If a Property with this name is already associated with the specified MetadataObject, then its value MUST be replaced with the specified value.

@Value: The value of the Property.

@ErrorCode: An error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-1	An error occurred while adding or replacing the specified Property.
-2	A MetadataObject with the specified <i>@MetadataObjectId</i> does not exist.
-3	The MetadataObject already contains the implementation-specific maximum allowed number of Properties.
A positive integer	A T-SQL SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.3 `proc_ar_BumpCacheInvalidationCounter`

The `proc_ar_BumpCacheInvalidationCounter` stored procedure is called to increment the cache version stamp for the given MetadataObject [MetadataObjectType](#), if the cache version stamp is not at an implementation-specific maximum value. If the cache version stamp is at the implementation-specific maximum value, the protocol server MUST set it to 0.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_BumpCacheInvalidationCounter (  
    @MetadataObjectType      nvarchar(250),  
    @ObjectCache             bit  
);
```

@MetadataObjectType: The type name of the MetadataObject whose cache version stamp will be incremented. The value MUST be a MetadataObjectType.

@ObjectCache: A bit that specifies the type of the cache version stamp to be affected. The value of this parameter MUST be listed in the following table.

Value	Description
0	The stored procedure MUST increment the version stamp of the relationship cache for the specified MetadataObject MetadataObjectType.
1	The stored procedure MUST increment the version stamp of the object cache for the specified MetadataObject MetadataObjectType.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.4 **proc_ar_ClearAccessControlEntriesForMetadataObject**

The **proc_ar_ClearAccessControlEntriesForMetadataObject** stored procedure is called to delete all access control entries associated with the specified MetadataObject.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_ClearAccessControlEntriesForMetadataObject(  
    @MetadataObjectId          int  
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject whose access control entries will be deleted.

Return Code Values: An integer that the protocol client MUST ignore.

Return Sets: MUST NOT return any result sets.

3.1.5.5 **proc_ar_CopyAccessControlEntriesForMetadataObjectId**

The **proc_ar_CopyAccessControlEntriesForMetadataObjectId** stored procedure is called to copy access control entries for a MetadataObject from another MetadataObject. If **@SourceMetadataObjectId** and **@DestinationMetadataObjectId** are equal, this stored procedure MUST make no changes. If **@SourceMetadataObjectId** and **@DestinationMetadataObjectId** are not equal, this stored procedure MUST delete all access control entries associated with the specified target MetadataObject and then copy the access control entries associated with the specified source MetadataObject.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CopyAccessControlEntriesForMetadataObjectId(  
    @SourceMetadataObjectId    int,  
    @DestinationMetadataObjectId int  
);
```

@SourceMetadataObjectId: The MetadataObjectId of the source MetadataObject from which the access control entries will be copied. The value MUST be an [Id](#).

@DestinationMetadataObjectId: The MetadataObjectId of the target MetadataObject to which the access control entries will be copied. The value MUST be an Id. The value MUST be an Id.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.6 **proc_ar_CreateAction**

The **proc_ar_CreateAction** stored procedure is called to create an Action in the specified Entity.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateAction(  

```



```

    @Name          nvarchar(50),
    @IsCached      bit,
    @EntityId      int,
    @Position      tinyint,
    @IsDisplayed   bit,
    @IsOpenedInNewWindow bit,
    @Icon          nvarchar(2080),
    @URL           nvarchar(2080),
    @CreatedId     int OUTPUT,
    @ErrorCode     int OUTPUT
);

```

@Name: The name of the Action. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Action is frequently used. The value MUST be an [IsCached](#).

@EntityId: The MetadataObjectId of the Entity that contains this Action. The value MUST be an [Id](#).

@Position: The order of this Action among the other Actions displayed in a user interface for this Entity. The value MUST be a [Position](#).

@IsDisplayed: A bit that provides a hint on whether the Action is displayed in the user interface presented to the user. The value MUST be an [IsDisplayed](#).

@IsOpenedInNewWindow: A bit that provides a hint on whether the results of executing the Action are displayed in a new window in the user interface presented to the user. The value MUST be an [IsOpenedInNewWindow](#).

@Icon: The URL of the icon associated with this Action. The value MUST be an [Icon](#).

@URL: The URL associated with this Action. The value MUST be an [URL](#).

@CreatedId: The identifier for the newly created Action. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created Action. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Entity already contains another Action with the specified <i>@Name</i> .
-3	The Entity already contains the implementation-specific maximum allowed number of Actions.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<7>](#)return zero or more result sets that the protocol client MUST ignore.

3.1.5.7 proc_ar_CreateActionParameter

The **proc_ar_CreateActionParameter** stored procedure is called to create an ActionParameter in the specified Action.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateActionParameter (
    @Name          nvarchar(50),
    @IsCached      bit,
    @ActionId      int,
    @Index         tinyint,
    @CreatedId     int OUTPUT,
    @ErrorCode     int OUTPUT
);
```

@Name: The name of the ActionParameter. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this ActionParameter is frequently used. The value MUST be an [IsCached](#).

@ActionId: The MetadataObjectId] of the Action that contains this ActionParameter. It MUST be [Id](#).

@Index: A value, indicating the position of this ActionParameter among the ActionParameters of the Action that contains this ActionParameter. It MUST be [Index](#).

@CreatedId: The MetadataObjectId of the newly created ActionParameter. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created ActionParameter. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors occurred.
-1	The Action already contains another ActionParameter with the specified <i>@Name</i> .
-3	The Action already contains the implementation-specific maximum allowed number of ActionParameters.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<8>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.8 proc_ar_CreateAssociation

The **proc_ar_CreateAssociation** stored procedure is called to create an Association. The stored procedure MUST copy the list of access control entries associated with the Entity that contains the Method that contains the Parameter that contains the TypeDescriptor whose MetadataObjectId is *@ReturnTypeDescriptorId* and associate them with the newly created Association.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_CreateAssociation (
    @Name                nvarchar(255),
    @IsCached            bit,
    @ReturnTypeDescriptorId int,
    @SourceEntityIds     nvarchar(255),
    @DestinationEntityId int,
    @CreatedId           int OUTPUT,
    @ErrorCode           int OUTPUT
);

```

@Name: The name of the Association. The value MUST be a [Name](#).

@IsCached: A bit that specifies where this Association is frequently used. The value MUST be an [IsCached](#).

@ReturnTypeDescriptorId: The MetadataObjectId of the ReturnTypeDescriptor. The value MUST be an [Id](#).

@SourceEntityIds: A comma-delimited list of MetadataObjectIds encoded as Unicode strings that represent the sources of the Association. For each value, there MUST exist an Entity with the specified MetadataObjectId. The value MUST NOT be NULL.

@DestinationEntityId: The MetadataObjectId of the Entity that represents the destination of the Association. The value MUST NOT be NULL.

@CreatedId: The identifier for the newly created Association. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created Association. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The LobSystem containing the Entity containing the Method containing the Parameter containing the TypeDescriptor with MetadataObjectId equal to <i>@ReturnTypeDescriptorId</i> already contains another Association with the specified <i>@Name</i> .
-3	Condition 1 MUST be true and condition 2 SHOULD <9> be true: Condition 1: The LobSystem containing the Entity containing the Method containing the Parameter containing the TypeDescriptor with MetadataObjectId equal to <i>@ReturnTypeDescriptorId</i> already contains the implementation-specific maximum allowed number of Associations Condition 2: The source Entities represented by <i>@SourceEntityIds</i> play the role of Source in less than an implementation-specific maximum number of Associations.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<10>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.9 `proc_ar_CreateEntity`

The `proc_ar_CreateEntity` stored procedure is called to create an Entity in the specified LobSystem. The stored procedure MUST copy the list of access control entries associated with the specified LobSystem and associate them with the newly created Entity.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateEntity] (  
    @Name                nvarchar(255),  
    @IsCached            bit,  
    @SystemId            int,  
    @EstimatedInstanceCount int,  
    @CreatedId           int OUTPUT,  
    @ErrorCode           int OUTPUT  
);
```

@Name: The name of the Entity. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Entity is frequently used. The value MUST be an [IsCached](#).

@SystemId: The MetadataObjectId of the LobSystem that contains this Entity. The value MUST be an [Id](#).

@EstimatedInstanceCount: The estimated number of instances of this Entity present within the LobSystemInstance. The value MUST be an [EstimatedInstanceCount](#). If this is NULL, the Entity MUST be created in the metadata store with a default value of 10000.

@CreatedId: The identifier for the newly created Entity. Upon return from this stored procedure with an `@ErrorCode` set to 0, this parameter MUST be set to the MetadataObjectId of the newly created Entity. Upon return from this stored procedure with an `@ErrorCode` set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The LobSystem already contains another Entity with the specified <code>@Name</code> .
-3	The LobSystem already contains the implementation-specific maximum allowed number of Entities.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<11>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.10 proc_ar_CreateFilterDescriptor

The **proc_ar_CreateFilterDescriptor** stored procedure is called to create a FilterDescriptor in the specified Method.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateFilterDescriptor (
    @Name                nvarchar(255),
    @IsCached            bit,
    @MethodId            int,
    @TypeName            nvarchar(255),
    @CreatedId           int OUTPUT,
    @ErrorCode           int OUTPUT
);
```

@Name: The name of the FilterDescriptor. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this FilterDescriptor is frequently used. The value MUST be an [IsCached](#).

@MethodId: The MetadataObjectId of the Method that contains this FilterDescriptor. The value MUST be an [Id](#).

@TypeName: The type name of the FilterDescriptor. The value MUST be a [FilterDescriptorTypeName](#).

@CreatedId: The identifier for the newly created FilterDescriptor. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created FilterDescriptor. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Method already contains another FilterDescriptor with the specified <i>@Name</i> .
-3	The Method already contains the implementation-specific maximum allowed number of FilterDescriptors.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<12>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.11 proc_ar_CreateIdentifier

The **proc_ar_CreateIdentifier** stored procedure is called to create an Identifier in the specified Entity.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_CreateIdentifier (
    @Name          nvarchar(255),
    @IsCached      bit,
    @EntityId      int,
    @TypeName      nvarchar(255),
    @CreatedId     int OUTPUT,
    @ErrorCode     int OUTPUT
);

```

@Name: The name of the Identifier. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Identifier is frequently used. The value MUST be an [IsCached](#).

@EntityId: The MetadataObjectId of the Entity that this contains this Identifier. The value MUST be an [Id](#).

@TypeName: The type name of the Identifier. The value MUST be an [IdentifierTypeName](#).

@CreatedId: The identifier for the newly created Identifier. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created Identifier. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Entity already contains another Identifier with the specified <i>@Name</i> .
-3	The Entity already contains the implementation-specific maximum number of Identifiers.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<13>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.12 proc_ar_CreateMethod

The **proc_ar_CreateMethod** stored procedure is called to create a Method in the specified DataClass. The stored procedure MUST copy the list of access control entries associated with the specified DataClass and associate them with the newly created Method.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_CreateMethod (
    @Name          nvarchar(255),
    @IsCached      bit,
    @ClassId       int,
    @IsStatic      bit,

```

```

        @CreatedId          int OUTPUT,
        @ErrorCode         int OUTPUT
    );

```

@Name: The name of the Method. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Method is frequently used. The value MUST be an [IsCached](#).

@ClassId: The MetadataObjectId of the DataClass that contains this Method. This MUST be [Id](#).

@IsStatic: A bit specifying whether the Method is associated with an EntityInstance. The value MUST be an [IsStatic](#).

@CreatedId: The identifier for the newly created Method. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created Method. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The DataClass already contains another Method with the specified <i>@Name</i> .
-3	The DataClass already contains the implementation-specific maximum allowed number of Methods.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<14>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.13 `proc_ar_CreateMethodInstance`

The `proc_ar_CreateMethodInstance` stored procedure is called to create a MethodInstance in the specified Method. The stored procedure MUST copy the list of access control entries associated with the parent DataClass of the specified Method and associate them with the newly created MethodInstance.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_CreateMethodInstance (
    @Name          nvarchar(255),
    @IsCached      bit,
    @MethodId      int,
    @ReturnTypeDescriptorId int,
    @Type          tinyint,
    @CreatedId     int OUTPUT,
    @ErrorCode     int OUTPUT
)

```

);

@Name: The name of the MethodInstance. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this MethodInstance is frequently used. The value MUST be an [IsCached](#).

@MethodId: The **MetadataObjectId of the** Method that contains this MethodInstance. The value MUST be an [Id](#).

@ReturnTypeDescriptorId: The MetadataObjectId of the ReturnTypeDescriptor. The value MUST be an Id.

@Type: The type of the MethodInstance. The value MUST be an [MethodInstanceType](#).

@CreatedId: The identifier for the newly created MethodInstance. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created MethodInstance. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors occurred.
-1	The DataClass of the specified Method already contains another MethodInstance with the specified <i>@Name</i> .
-3	The specified Method already contains the implementation-specific maximum allowed number of MethodInstances.
-200	<i>@Type</i> equals 1 and Entity or DataClass of the Method already contains another MethodInstance of type Finder.
-201	<i>@Type</i> equals 2 and Entity or DataClass of the Method already contains another MethodInstance of type SpecificFinder.
-202	<i>@Type</i> equals 5 and Entity or DataClass of the Method already contains another MethodInstance of type IdEnumerator.
-203	MetadataObjectId of the Method containing the Parameter of the TypeDescriptor with MetadataObjectId equal to <i>@ReturnTypeDescriptorId</i> is not equal to <i>@MethodId</i> .
-204	The Parameter that contains the specified ReturnTypeDescriptor cannot have a Direction of 1.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<15>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.14 proc_ar_CreateParameter

The **proc_ar_CreateParameter** stored procedure is called to create a Parameter contained by the Method identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateParameter (  
    @Name                nvarchar(255),  
    @IsCached            bit,  
    @MethodId            int,  
    @Direction           tinyint,  
    @TypeReflectorTypeName nvarchar(255),  
    @CreatedId           int OUTPUT,  
    @ErrorCode           int OUTPUT  
);
```

@Name: The programmatic name of the Parameter. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Parameter is frequently used. The value MUST be an [IsCached](#).

@MethodId: The MetadataObjectId of the Method that contains this Parameter. The value MUST be an [Id](#).

@Direction: The direction in which this Parameter is passed to the LOB System while calling the Method. The value MUST be a [Direction](#).

@TypeReflectorTypeName: The type name of the TypeReflector that will be used to resolve the native type of this parameter. The value MUST be a [TypeReflectorTypeName](#).

@CreatedId: The identifier for the newly created Parameter. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created Parameter. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors occurred.
-1	The Method already has a Parameter with the specified <i>@Name</i> .
-3	The Method already contains the implementation-specific maximum allowed number of Parameters.
-100	<i>@Direction</i> is set to 4 and the Method already has a return Parameter.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<16>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.15 proc_ar_CreateSystem

The **proc_ar_CreateSystem** stored procedure is called to create a LobSystem in the **ApplicationRegistry**<17>. It MUST copy the list of access control entries associated with the ApplicationRegistry and associate them with the newly created LobSystem.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateSystem] (  
    @Name nvarchar(255),  
    @IsCached bit,  
    @SystemUtilityTypeName nvarchar(255),  
    @ConnectionManagerTypeName nvarchar(255),  
    @EntityInstanceTypeName nvarchar(255),  
    @CreatedId int OUTPUT,  
    @ErrorCode int OUTPUT  
);
```

@Name: The name of the LobSystem. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this LobSystem is frequently used. The value MUST be an [IsCached](#).

@SystemUtilityTypeName: The name of the System Utility that will be used to execute the Methods in this LobSystem. The value MUST be a [SystemUtilityTypeName](#).

@ConnectionManagerTypeName: The name of the connection manager that will be used while connecting to this LobSystem. The value MUST be a [ConnectionManagerTypeName](#).

@EntityInstanceTypeName: The name of the unit of implementation-specific <18> business logic (2) that will be used to create the objects that will carry EntityInstance data to client applications. The value MUST be an [EntityInstanceTypeName](#).

@CreatedId: The identifier for the newly created LobSystem. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created LobSystem. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors occurred.
-1	The ApplicationRegistry already contains another LobSystem with the specified <i>@Name</i> .
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY <19> return zero or more result sets that the protocol client MUST ignore.

3.1.5.16 proc_ar_CreateSystemInstance

The **proc_ar_CreateSystemInstance** stored procedure is called to create a LobSystemInstance in the specified LobSystem. It MUST copy the list of access control entries associated with the LobSystem and associate them with the newly created LobSystemInstance.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateSystemInstance] (  
    @Name                nvarchar(255),  
    @IsCached            bit,  
    @SystemId            int,  
    @CreatedId           int OUTPUT,  
    @ErrorCode           int OUTPUT  
);
```

@Name: The name of the LobSystemInstance. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this LobSystemInstance is frequently used. The value MUST be an [IsCached](#).

@SystemId: The MetadataObjectId of the LobSystem that contains this LobSystemInstance. The value MUST be an [Id](#).

@CreatedId: The identifier for the newly created LobSystemInstance. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created LobSystemInstance. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors occurred.
-1	A LobSystemInstance with MetadataObject name equal to <i>@Name</i> already exists.
-3	The LobSystem already contains the implementation-specific maximum allowed number of LobSystemInstances.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<20>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.17 proc_ar_CreateTypeDescriptor

The **proc_ar_CreateTypeDescriptor** stored procedure is called to create a TypeDescriptor.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_CreateTypeDescriptor (  
    @Name                nvarchar(255),
```

```

@IsCached          bit,
@ParameterId      int,
@ParentTypeDescriptorId int,
@TypeName         nvarchar(255),
@IdentifierId     int,
@FilterDescriptorId int,
@IsCollection     bit,
@CreatedId        int OUTPUT,
@ErrorCode        int OUTPUT
);

```

@Name: The name of the TypeDescriptor. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this TypeDescriptor is frequently used. The value MUST be an [IsCached](#).

@ParameterId: The MetadataObjectId of the Parameter that contains this TypeDescriptor. The value MUST be an [Id](#).

@ParentTypeDescriptorId: The MetadataObjectId of the parent **TypeDescriptor** that contains this TypeDescriptor. The value MUST be an [Id](#).

@TypeName: The programmatic name of the data type that is represented by this TypeDescriptor. The value MUST be a [TypeDescriptorTypeName](#).

@IdentifierId: The MetadataObjectId of the **Identifier** referenced by this TypeDescriptor. The value MUST be an [Id](#).

@FilterDescriptorId: The MetadataObjectId of the **FilterDescriptor** associated with this TypeDescriptor. The value MUST be an [Id](#).

@IsCollection: A bit that specifies whether this TypeDescriptor is to be interpreted by protocol clients as a collection of native LOB System data structures. The value MUST be an [IsCollection](#).

@CreatedId: The identifier for the newly created TypeDescriptor. Upon return from this stored procedure with an *@ErrorCode* set to 0, this parameter MUST be set to the MetadataObjectId of the newly created TypeDescriptor. Upon return from this stored procedure with an *@ErrorCode* set to a value other than 0, this parameter is set to a value that MUST be ignored.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors occurred.
-1	The TypeDescriptor identified by <i>@ParentTypeDescriptor</i> already contains a child TypeDescriptor with the specified <i>@Name</i> .
-3	At least one of the following two statements is true: <ul style="list-style-type: none"> ▪ <i>@ParentTypeDescriptorId</i> is not null and a TypeDescriptor with MetadataObjectId equal to <i>@ParentTypeDescriptorId</i> already contains the implementation-specific maximum allowed number of TypeDescriptors. ▪ <i>@FilterDescriptorId</i> is not null and a FilterDescriptor with MetadataObjectId equal to <i>@FilterDescriptorId</i> already contains the implementation-specific maximum allowed number

Value	Description
	of associated TypeDescriptors.
-302	The @ParentTypeDescriptorId is equal to null and Parameter already has a root TypeDescriptor.
-303	The Parameter and FilterDescriptor with MetadataObjectId equal to @FilterDescriptorId don't belong to the same Method.
-305	@IsCollection equals 1 and the data type represented by the TypeDescriptor with MetadataObjectId equal to @ParentTypeDescriptorId is interpreted as a collection. A child TypeDescriptor of a parent TypeDescriptor that represents a collection MUST NOT be a collection.
-306	The data type specified by @ParentTypeDescriptorId is interpreted as a collection and already contains a child TypeDescriptor. A TypeDescriptor that represents a collection MUST NOT have more than one child.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<21>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.18 `proc_ar_DeleteActionById`

The `proc_ar_DeleteActionById` stored procedure is called to delete the specified Action along with its Properties, localized names, access control entries and ActionParameters.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_DeleteActionById (
    @Id                int,
    @Version           int,
    @ErrorCode         int OUTPUT
);

```

@Id: The MetadataObjectId of the Action that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this Action at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	An Action with the specified MetadataObjectId does not exist.
-6	An Action with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Action.
A positive	A T-SQL error code.

Value	Description
integer	

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<22>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.19 `proc_ar_DeleteActionParameterById`

The `proc_ar_DeleteActionParameterById` stored procedure is called to delete the `ActionParameter` identified by its given `MetadataObjectId` along with its Properties, localized names and access control entries.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteActionParameterById (
    @Id                int,
    @Version            int,
    @ErrorCode          int OUTPUT
);
```

@Id: The identifier for the `ActionParameter` that is to be deleted. The value MUST be an [Id](#).

@Version: The object version of this `ActionParameter` at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	An <code>ActionParameter</code> with <code>MetadataObjectId</code> equal to <code>@Id</code> doesn't exist.
-6	An <code>ActionParameter</code> with <code>MetadataObjectId</code> equal to <code>@Id</code> has been updated by a context other than the one that it has been currently read by. This happens when the value of <code>@Version</code> does not match with the version for the <code>ActionParameter</code> .
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<23>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.20 `proc_ar_DeleteAssociationById`

The `proc_ar_DeleteAssociationById` stored procedure is called to delete the `Association` identified by its given `MetadataObjectId` along with its Properties, localized names and access control entries.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteAssociationById (
    @Id                int,
```

```

    @Version          int,
    @ErrorCode        int OUTPUT
);

```

@Id: The identifier for the Association that is to be deleted. The value MUST be an [Id](#).

@Version: The object version of this Association at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	An Association with MetadataObjectId equal to <i>@Id</i> doesn't exist.
-6	An Association with MetadataObjectId equal to <i>@Id</i> has been updated by a context other than the one that it has been currently read by. This happens when the value of <i>@Version</i> does not match with the version of the Association.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<24>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.21 `proc_ar_DeleteDefaultValue`

The `proc_ar_DeleteDefaultValue` stored procedure is called to delete the **DefaultValue** identified by the specified TypeDescriptor and MethodInstance.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_DeleteDefaultValue (
    @TypeDescriptorId    int,
    @MethodInstanceId    int,
    @ErrorCode            int OUTPUT
);

```

@TypeDescriptorId: The MetadataObjectId of the TypeDescriptor associated with the DefaultValue. The value MUST be an [Id](#).

@MethodInstanceId: The MetadataObjectId of the MethodInstance associated with the DefaultValue. The value MUST be an Id.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	At least one of the following two statements is true:

Value	Description
	<ul style="list-style-type: none"> ▪ A TypeDescriptor with MetadataObjectId equal to <i>@TypeDescriptorId</i> doesn't exist. ▪ A MethodInstance with MetadataObjectId equal to <i>@MethodInstanceId</i> doesn't exist.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY <25> return zero or more result sets that the protocol client MUST ignore.

3.1.5.22 **proc_ar_DeleteEntityById**

The **proc_ar_DeleteEntityById** stored procedure is called to delete the Entity identified by the specified MetadataObjectId along with its Properties, localized names, and access control entries.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_DeleteEntityById (
    @Id                int,
    @Version           int,
    @ErrorCode         int OUTPUT
);

```

@Id: The MetadataObjectId of the Entity that will be updated. The value MUST be an [Id](#).

@Version: The object version of this Entity at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	An Entity with the specified <i>@Id</i> does not exist.
-5	An Entity with the specified <i>@Id</i> is referenced as a source or destination in any Association, or contains at least one of the following child objects: <ul style="list-style-type: none"> ▪ Action. ▪ Method. ▪ Identifier.
-6	An Entity with the specified <i>@Id</i> has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Entity.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<26>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.23 `proc_ar_DeleteFilterDescriptorById`

The `proc_ar_DeleteFilterDescriptorById` stored procedure is called to delete the `FilterDescriptor` identified by the specified `MetadataObjectId` along with its Properties, localized names, and access control entries.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteFilterDescriptorById (  
    @Id                int,  
    @Version           int,  
    @ErrorCode         int OUTPUT  
);
```

@Id: The `MetadataObjectId` of the `FilterDescriptor` that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this `FilterDescriptor` at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	A <code>FilterDescriptor</code> with the specified <code>MetadataObjectId</code> does not exist.
-6	A <code>FilterDescriptor</code> with the specified <code>MetadataObjectId</code> has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the <code>FilterDescriptor</code> .
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<27>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.24 `proc_ar_DeleteIdentifierById`

The `proc_ar_DeleteIdentifierById` stored procedure is called to delete the `Identifier` identified by the specified `MetadataObjectId` along with its Properties, localized names, and access control entries. After a successful deletion, the ordinal number attribute of all `Identifiers` MUST be normalized for `Identifiers` that are contained by the same `Entity` that contained the deleted `Identifier`. After normalization, the ordinal number of all these `Identifiers` MUST be renumbered starting from 0, incrementing by 1 and preserving the original ordering. During this renumbering, the version attribute of all these `Identifiers` SHOULD [<28>](#) be updated.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteIdentifierById (  
    @Id                int,
```

```

    @Version          int,
    @ErrorCode        int OUTPUT
);

```

@Id: The MetadataObjectId of the Identifier] that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this Identifier at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	An Identifier with the specified MetadataObjectId does not exist.
-6	An Identifier with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Identifier.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<29>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.25 `proc_ar_DeleteLocalizedNameForMetadataObjectByLCID`

The `proc_ar_DeleteLocalizedNameForMetadataObjectByLCID` stored procedure is called to delete a localized name contained by the MetadataObject for a given LCID.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_DeleteLocalizedNameForMetadataObjectByLCID (
    @MetadataObjectId    int,
    @LCID                int,
    @ErrorCode           int OUTPUT
);

```

@MetadataObjectId: The MetadataObjectId of the MetadataObject that contains the localized name to be deleted. The value MUST be an [Id](#).

@LCID: The LCID of the localized name to be deleted.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	A localized name for the given @LCID doesn't exist or exists more than once for the MetadataObject with MetadataObjectId equal to @MetadataObjectId.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST NOT return any result sets.

3.1.5.26 `proc_ar_DeleteLocalizedNamesByMetadataObjectId`

The `proc_ar_DeleteLocalizedNamesByMetadataObjectId` stored procedure is called to delete all localized names of the MetadataObject identified by its specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteLocalizedNamesByMetadataObjectId (  
    @MetadataObjectId      int,  
    @ErrorCode             int OUTPUT  
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject that owns the localized names to be deleted.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST NOT return any result sets.

3.1.5.27 `proc_ar_DeleteMethodById`

The `proc_ar_DeleteMethodById` stored procedure is called to delete the Method identified by the specified MetadataObjectId along with its Properties, localized names, and access control entries.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteMethodById (  
    @Id                    int,  
    @Version               int,  
    @ErrorCode             int OUTPUT  
);
```

@Id: The MetadataObjectId of the Method that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this Method at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	A Method with the specified MetadataObjectId does not exist.
-5	A Method with the specified MetadataObjectId contains at least one of the following child objects: <ul style="list-style-type: none"> ▪ FilterDescriptor ▪ MethodInstance ▪ Parameter
-6	A Method with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Method.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<30>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.28 `proc_ar_DeleteMethodInstanceById`

The `proc_ar_DeleteMethodInstanceById` stored procedure is called to delete the MethodInstance identified by the specified MetadataObjectId along with its Properties, localized names, and access control entries. It MUST also delete any DefaultValues associated with the MethodInstance identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_DeleteMethodInstanceById (
    @Id          int,
    @Version     int,
    @ErrorCode   int OUTPUT
);

```

@Id: The MetadataObjectId of the MethodInstance that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this MethodInstance at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	A MethodInstance with the specified MetadataObjectId does not exist.
-6	A MethodInstance with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified

Value	Description
	does not match with the current version of the MethodInstance.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<31>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.29 `proc_ar_DeleteParameterById`

The `proc_ar_DeleteParameterById` stored procedure is called to delete the Parameter identified by its given MetadataObjectId along with its Properties, localized names and access control entries. After a successful deletion, the ordinal number attribute of all Parameters MUST be normalized for Parameters that are contained by the same Method that contained the deleted Parameter. After normalization, the ordinal number of all these Parameters MUST be renumbered starting from 0, incrementing by 1 and preserving the original ordering. During this renumbering, the version attribute of all these Parameters SHOULD [<32>](#) be updated.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteParameterById (
    @Id                int,
    @Version            int,
    @ErrorCode          int OUTPUT
);
```

@Id: The MetadataObjectId of the Parameter to be deleted. The value MUST be an [Id](#).

@Version: The object version of this Parameter at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	A Parameter with MetadataObjectId equal to <code>@Id</code> doesn't exist.
-5	One or more TypeDescriptors with ParameterId equal to <code>@Id</code> already exist.
-6	A Parameter with MetadataObjectId equal to <code>@Id</code> has been updated by a context other than the one that it has been currently read by. This happens when the value of <code>@Version</code> does not match the version for the Parameter.
A positive integer	A T-SQL error.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<33>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.30 proc_ar_DeletePropertiesById

The **proc_ar_DeletePropertiesById** stored procedure is called to delete all Properties contained by the MetadataObject identified by its given MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeletePropertiesById (
    @MetadataObjectId      int,
    @ErrorCode              int OUTPUT
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject that contains the Properties to be deleted. The value MUST be an [Id](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
A positive integer	A T-SQL error.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST NOT return any result sets.

3.1.5.31 proc_ar_DeletePropertyForMetadataObjectId

The **proc_ar_DeletePropertyForMetadataObjectId** stored procedure is called to delete the Property with a given programmatic name and contained by a MetadataObject identified by its given MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeletePropertyForMetadataObjectId (
    @MetadataObjectId      int,
    @Name                  nvarchar(255),
    @ErrorCode              int OUTPUT
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject that contains the Property to be deleted. The value MUST be an [Id](#).

@Name: The name of the Property. The value MUST be a [Name](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.

Value	Description
-2	A Property with name equal to <i>@Name</i> doesn't exist, or exists more than once for the MetadataObject with a MetadataObjectId equal to <i>@MetadataObjectId</i> .

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST NOT return any result sets.

3.1.5.32 *proc_ar_DeleteSystemById*

The ***proc_ar_DeleteSystemById*** stored procedure is called to delete the LobSystem identified by the specified MetadataObjectId along with its Properties, localized names, and access control entries.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteSystemById (
    @Id                int,
    @Version           int,
    @ErrorCode         int OUTPUT
);
```

@Id: The MetadataObjectId of the LobSystem that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this LobSystem at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	A LobSystem with the specified MetadataObjectId does not exist.
-5	A LobSystem with the specified MetadataObjectId contains at least one of the following child objects: <ul style="list-style-type: none"> ▪ DataClasses ▪ LobSystemInstances
-6	A LobSystem with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the LobSystem.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<34>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.33 proc_ar_DeleteSystemInstanceById

The **proc_ar_DeleteSystemInstanceById** stored procedure is called to delete the LobSystemInstance identified by the specific MetadataObjectId along with its access control entries.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteSystemInstanceById (
    @Id                int,
    @Version            int,
    @ErrorCode          int OUTPUT
);
```

@Id: The MetadataObjectId of the LobSystemInstance that is to be deleted. The value MUST be an [Id](#).

@Version: The object version of this LobSystemInstance at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	An LobSystemInstance with the specified MetadataObjectId does not exist.
-6	An LobSystemInstance with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the LobSystemInstance.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<35>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.34 proc_ar_DeleteTypeDescriptorById

The **proc_ar_DeleteTypeDescriptorById** stored procedure is called to delete the TypeDescriptor identified by the specified MetadataObjectId along with its Properties, localized names, access control entries and all its child TypeDescriptors, recursively.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_DeleteTypeDescriptorById (
    @Id                int,
    @Version            int,
    @ErrorCode          int OUTPUT
);
```

@Id: The MetadataObjectId of the TypeDescriptor that will be deleted. The value MUST be an [Id](#).

@Version: The object version of this TypeDescriptor at the time it was last read.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	A TypeDescriptor with the specified MetadataObjectId does not exist..
-5	A MethodInstance refers to the TypeDescriptor identified by the specified MetadataObjectId as its ReturnTypeDescriptor.
-6	A TypeDescriptor with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the TypeDescriptor.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MAY [<36>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.35 **proc_ar_EnsureApplicationRegistryExists**

The **proc_ar_EnsureApplicationRegistryExists** stored procedure is called to verify that an ApplicationRegistry exists. When not found, the stored procedure MUST create a new ApplicationRegistry.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_EnsureApplicationRegistryExists ();
```

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.36 **proc_ar_GetAccessControlEntriesForMetadataObject**

The **proc_ar_GetAccessControlEntriesForMetadataObject** stored procedure is called to retrieve access control entries for a MetadataObject with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetAccessControlEntriesForMetadataObject (  
    @MetadataObjectId      int,  
    @ErrorCode              int OUTPUT  
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-2	Object not found.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: If @ErrorCode is set to -2, this stored procedure MUST NOT return any result sets. Otherwise this stored procedure MUST return the following result set:

3.1.5.36.1 Access Control Entry Result Set

The **Access Control Entry** Result Set returns information about access control entries that authorize what may be done with the MetadataObject it is associated with. Each row in the result set contains all the attributes of a single access control entry. The result set MUST have zero or more rows.

The T-SQL syntax for the result set is as follows:

```
MetadataObjectId      int,
IdentityName          nvarchar(255),
DisplayName            nvarchar(255),
RawSid                varbinary(512),
Rights                bigint;
```

MetadataObjectId: The MetadataObjectId of the MetadataObject that the access control entry is associated with.

IdentityName: The programmatic name of the security principal (2).

DisplayName: The name of the security principal (2) used for display purposes.

RawSid: The **SID**, if the security principal (2) is a Windows security principal. If the security principal (2) is not a windows security principal, the value MUST be NULL.

Rights: The permissions available to the security principal (2) for the MetadataObject identifier by the MetadataObjectId. It MUST be [MetadataRights](#).

3.1.5.37 proc_ar_GetActionById

The **proc_ar_GetActionById** stored procedure is called to retrieve Action information for the Action with a specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetActionById (
    @MetadataObjectId      int
);
```

@MetadataObjectId: The MetadataObjectId of the Action. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.37.1 Action Result Set

See [Action result set](#). The result set MUST contain zero or one row.

3.1.5.38 proc_ar_GetActionParameterById

The **proc_ar_GetActionParameterById** stored procedure is called to retrieve ActionParameter data for the ActionParameter with a specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetActionParameterById (
    @MetadataObjectId      int
);
```

@MetadataObjectId: The MetadataObjectId of the ActionParameter.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set:

3.1.5.38.1 ActionParameter Result Set

See [ActionParameter result set](#). The result set MUST contain zero or one row.

3.1.5.39 proc_ar_GetActionParametersForActionWithCount

The **proc_ar_GetActionParametersForActionWithCount** stored procedure is called to retrieve the ActionParameters contained by the Action with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetActionParametersForActionWithCount (
    @ActionId      int
);
```

@ActionId: The MetadataObjectId of the Action. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets.

3.1.5.39.1 Count Result Set

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.39.2 ActionParameter Result Set

See [ActionParameter Result Set](#). The result set MUST contain zero or more rows.

3.1.5.40 **proc_ar_GetActionsForEntityWithCount**

The **proc_ar_GetActionsForEntityWithCount** stored procedure is called to retrieve the Actions contained by the Entity with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetActionsForEntityWithCount (
    @EntityId          int
);
```

@EntityId: The MetadataObjectId of the Entity. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets.

3.1.5.40.1 **Count Result Set**

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.40.2 **Action Result Set**

See [Action Result Set](#). The result set MUST contain zero or more rows.

3.1.5.41 **proc_ar_GetAllLocalizedNamesForMetadataObjectWithCount**

The **proc_ar_GetAllLocalizedNamesForMetadataObjectWithCount** stored procedure is called to retrieve all localized names of a MetadataObject identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE [proc_ar_GetAllLocalizedNamesForMetadataObjectWithCount] (
    @MetadataObjectId int
);
```

@MetadataObjectId: The Id of the MetadataObject. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets.

3.1.5.41.1 **Count Result Set**

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.41.2 **Localized Name Result Set**

The Localized Name Result Set contains localized names. Each row in the result set contains a single localized name of a MetadataObject in a specific **locale**.

The T-SQL syntax for the result set is as follows:

Id	int,
LCID	int,
[Localized name]	nvarchar(255),
MetadataObjectId	int;

Id: The MetadataObjectId of the localized name that is returned. The value MUST be an [Id](#).

LCID: The **LCID** corresponding to the returned localized name.

Localized name: The localized name of the specified MetadataObject corresponding to the returned LCID.

MetadataObjectId: The MetadataObjectId of the MetadataObject. The value MUST be an Id.

The result set MUST contain zero or more rows.

3.1.5.42 **proc_ar_GetAllSystemInstancesLikeNameWithCount**

The **proc_ar_GetAllSystemInstancesLikeNameWithCount** stored procedure is called to retrieve the count and the LobSystemInstances that satisfy either one of the following constraints:

- LobSystemInstances whose name matches the pattern given by *@SystemInstanceName*.
- LobSystemInstances whose localized names matches the pattern given by *@SystemInstanceName*, and for which the LCID of the same localized name matches either of the following constraints:
 - is equal with the value given by *@LCID* or
 - is equal to zero.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE [proc_ar_GetAllSystemInstancesLikeNameWithCount] (
    @SystemInstanceName    nvarchar(255),
    @LCID                  int
);
```

@SystemInstanceName: The string that specifies the name pattern of the LobSystemInstances to be returned. The characters in this string MUST be in upper case. It can include wildcard characters. For example, setting the *@MetadataObjectName* as "A%" will make this stored procedure return only the LobSystemInstances with names beginning with either "A" or "a".

@LCID: The LCID used to restrict which localized names of the LobSystemInstances to consider.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets:

3.1.5.42.1 **Count Result Set**

See [Count result set](#). The result set MUST return one row.

3.1.5.42.2 System Instance Result Set

See [Entity result set](#). The result set MUST return zero or more rows.

3.1.5.43 proc_ar_GetAllSystemInstancesWithCount

The **proc_ar_GetAllSystemInstancesWithCount** stored procedure is called to retrieve all LobSystemInstances, along with the count of such LobSystemInstances.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetAllSystemInstancesWithCount();
```

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.43.1 Count Result Set

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.43.2 System Instance Result Set

See [System Instance Result Set](#). The result set MUST contain zero or more rows.

3.1.5.44 proc_ar_GetAllSystemsWithCount

The **proc_ar_GetAllSystemsWithCount** stored procedure is called to retrieve the count and details of all LobSystems.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetAllSystemsWithCount();
```

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets:

3.1.5.44.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.44.2 System Result Set

See [System result set](#). The result set MUST contain zero or more rows.

3.1.5.45 proc_ar_GetAssociationById

The **proc_ar_GetAssociationById** stored procedure is called to retrieve the Association identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_GetAssociationById (
    @MetadataObjectId      int
);

```

@MetadataObjectId: The MetadataObjectId of the Association that will be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Set: MUST return the following result set:

3.1.5.45.1 Association Result Set

See [Association Result Set](#). The result set MUST contain zero or one row.

3.1.5.46 proc_ar_GetAssociationByName

The **proc_ar_GetAssociationByName** stored procedure is called to retrieve the Association with the specified name contained by the specified LobSystem identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_GetAssociationByName (
    @AssociationName      nvarchar(255),
    @SystemId             int
);

```

@AssociationName: The programmatic name of the Association that will be retrieved. The value MUST be a [Name](#).

@SystemId: The MetadataObjectId of the LobSystem that contains this Association. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set:

3.1.5.46.1 Association Result Set

See [Association Result Set](#). The result set MUST contain zero or one row.

3.1.5.47 proc_ar_GetAssociationsForDataClassWithCount

The **proc_ar_GetAssociationsForDataClassWithCount** stored procedure is invoked to retrieve the count and details of all Associations contained by all Methods contained by the specified DataClass. The stored procedure MUST return all Associations for all Methods of a DataClass, but no MethodInstances that are not Associations.

The T-SQL syntax is as follows:

```

PROCEDURE proc_ar_GetAssociationsForDataClassWithCount (
    @ClassId              int
);

```

);

@ClassId: the MetadataObjectId for the DataClass.

Return Code Values: An integer that the protocol client MUST be ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.47.1 Count Result Set

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.47.2 Association Result Set

See [Association Result Set](#). The result set MUST contain zero or more rows.

3.1.5.48 proc_ar_GetAssociationsForEntityAndRoleWithCount

The **proc_ar_GetAssociationsForEntityAndRoleWithCount** stored procedure is called to retrieve the Associations that reference the specified Entity as an Association source or destination.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetAssociationsForEntityAndRoleWithCount (  
    @EntityId          int,  
    @EntityRole        bit  
);
```

@EntityId: The MetadataObjectId of the Entity. The value MUST be an [Id](#).

@EntityRole: A bit that specifies whether *@EntityId* represents a source or a destination Entity. The value of this parameter MUST be listed in the following table.

Value	Description
0	Source Entity.
1	Destination Entity.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.48.1 Count Result Set

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.48.2 Association Result Set

See [Association Result Set](#). The result set MUST contain zero or more rows.

3.1.5.49 **proc_ar_GetAssociationsForMethodWithCount**

The **proc_ar_GetAssociationsForMethodWithCount** stored procedure is called to retrieve the count and details of all Associations contained by the Method. MethodInstances that are not Associations MUST NOT be returned.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetAssociationsForMethodWithCount (
    @MethodId          int
);
```

@MethodId: The MetadataObjectId of the Method object. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets:

3.1.5.49.1 **Count Result Set**

See [Count result set](#). The result set MUST contain one row.

3.1.5.49.2 **Association Result Set**

See [Association result set](#). The result set MUST contain zero or more rows.

3.1.5.50 **proc_ar_GetCacheInvalidationCountersWithCount**

The **proc_ar_GetCacheInvalidationCountersWithCount** stored procedure is called to retrieve current cache version stamp information along with the count of version stamp for the cache.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetCacheInvalidationCountersWithCount();
```

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result sets.

3.1.5.50.1 **Count Result Set**

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.50.2 **Cache Version Stamps Result Set**

The **Cache Version Stamps** Result Set returns information about the version stamps for the cache. Each row in the result set MUST represent object cache version stamp and relationship cache version stamp for a given [MetadataObjectType](#). The version stamps for the cache MUST be in ascending order of their MetadataObjectType attribute.

The Cache Version Stamps Result Set is defined using T-SQL syntax, as follows:

```
MetadataObjectType          nvarchar(255),
```

```
ObjectCacheCounter      int,  
RelationshipCacheCounter int;
```

MetadataObjectType: The type of the MetadataObject. It MUST be MetadataObjectType.

ObjectCacheCounter: The Object cache version stamp for the MetadataObjectType.

RelationshipCacheCounter: The relationship cache version stamp for the MetadataObjectType.

The result set MUST contain zero or more rows.

3.1.5.51 **proc_ar_GetChildTypeDescriptorsForTypeDescriptorWithCount**

The **proc_ar_GetChildTypeDescriptorsForTypeDescriptorWithCount** stored procedure is called to retrieve the count and details of child TypeDescriptors for the specified TypeDescriptor.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetChildTypeDescriptorsForTypeDescriptorWithCount (  
    @ParentTypeDescriptorId      int  
)  
;
```

@ParentTypeDescriptorId: The MetadataObjectId for the parent TypeDescriptor. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.51.1 **Count Result Set**

See [Count result set](#). The result set MUST contain one row.

3.1.5.51.2 **TypeDescriptor Result Set**

See [Type Descriptor result set](#). The result set MUST contain zero or more rows.

3.1.5.52 **proc_ar_GetDataClassById**

The **proc_ar_GetDataClassById** stored procedure is called to retrieve the DataClass identified by the specified MetadataObjectId.

The T-SQL syntax for this stored procedure is as follows:

```
PROCEDURE proc_ar_GetDataClassById (  
    @MetadataObjectId      int  
)  
;
```

@MetadataObjectId: The MetadataObjectId of the DataClass to be returned. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Set: MUST return a single result set.

3.1.5.52.1 DataClass Result Set

See [DataClass result set](#). The result set MUST contain zero or one row.

3.1.5.53 proc_ar_GetDataClassesForSystemWithCount

The **proc_ar_GetDataClassesForSystemWithCount** stored procedure is called to retrieve the count and details of DataClasses contained by the specified LobSystem.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetDataClassesForSystemWithCount (
    @SystemId          int
);
```

@SystemId: The MetadataObjectId for the LobSystem object. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.53.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.53.2 DataClass Result Set

See [DataClass result set](#). The result set MUST contain zero or more rows.

3.1.5.54 proc_ar_GetDefaultValuesForTypeDescriptor

The **proc_ar_GetDefaultValuesForTypeDescriptor** stored procedure is called to retrieve DefaultValues associated with the TypeDescriptor identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetDefaultValuesForTypeDescriptor (
    @TypeDescriptorId  int,
    @ErrorCode         int OUTPUT
);
```

@TypeDescriptorId: The MetadataObjectId for the TypeDescriptor object. The value MUST be an [Id](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	Object not found. The <i>@TypeDescriptorId</i> parameter does not identify a valid TypeDescriptor.

Return Code Values: An integer which the protocol client MUST ignore.

Result Sets: MUST return a single result set.

3.1.5.54.1 DefaultValues Result Set

The **Default Values** Result Set returns the DefaultValues of the TypeDescriptor. It MUST contain zero or more rows. The Result Set MUST contain zero rows if the specified TypeDescriptor does not have any DefaultValues defined for any of the MethodInstances using it. DefaultValues.

The T-SQL syntax for this result set is as follows:

```
Id                int,  
Value             sql_variant,  
TypeDescriptorId int,  
MethodInstanceId int,  
MethodInstanceName nvarchar(255);
```

Id: An implementation-specific identifier for the DefaultValue.

Value: The value of the DefaultValue.

TypeDescriptorId: The MetadataObjectId of the TypeDescriptor with which the DefaultValue is associated. It MUST be an [Id](#).

MethodInstanceId: The MetadataObjectId of the MethodInstance with which the DefaultValue is associated. It MUST be an Id.

MethodInstanceName: The programmatic name of the MethodInstance identified by the specified MetadataObjectId equal to *@MethodInstanceId*.

3.1.5.55 proc_ar_GetDependentEntitiesForEntity

The **proc_ar_GetDependentEntitiesForEntity** stored procedure is called to retrieve MetadataObjectIds for the Entities that are referenced by the specified Entity. Whenever a Method for the specified Entity has a Parameter with a TypeDescriptor that refers to another Entity, that Entity information MUST be included. The information about the original Entity MUST be excluded.

The T-SQL syntax for this stored procedure is as follows:

```
PROCEDURE proc_ar_GetDependentEntitiesForEntity (  
    @EntityId int  
);
```

@EntityId: The MetadataObjectId for the entity object. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return a single result set.

3.1.5.55.1 EntityId Result Set

The **EntityId** Result Set returns MetadataObjectIds for a set of Entities. The result set can have zero or more rows. There MUST be no duplicate MetadataObjectIds in the result set.

The T-SQL syntax for the Result Set is as follows:

```
EntityId          int;
```

EntityId: The MetadataObjectId of an Entity.

3.1.5.56 proc_ar_GetEntitiesForAssociationAndRoleWithCount

The **proc_ar_GetEntitiesForAssociationAndRoleWithCount** stored procedure is invoked to retrieve the count and details of Entities representing an Association source or destination for the specified Association.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetEntitiesForAssociationAndRoleWithCount (  
    @AssociationId      int,  
    @EntityRole        bit  
);
```

@AssociationId: The MetadataObjectId of the Association. The value MUST be an [Id](#).

@EntityRole: A bit that specifies whether to return Entities representing an Association source or destination. The value of this parameter MUST be listed in the following table.

Value	Description
0	Source Entity.
1	Destination Entity.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.56.1 Count Result Set

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.56.2 Entity Result Set

See [Entity Result Set](#). The result set MUST contain zero or more rows.

3.1.5.57 proc_ar_GetEntitiesForSystemLikeNameWithCount

The **proc_ar_GetEntitiesForSystemLikeNameWithCount** stored procedure is called to retrieve the count and the Entities that satisfy all of the following constraints:

- Entities contained by LobSystem with the specified MetadataObjectId.

- Entities whose name matches the given pattern.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetEntitiesForSystemLikeNameWithCount (  
    @MetadataObjectName    nvarchar(255),  
    @SystemId              int  
);
```

@MetadataObjectName: The string that defines the name pattern of the Entities to be returned. The characters in this string MUST be in upper case. It can include wildcard characters. For example, setting the *@MetadataObjectName* parameter as "A%" will make this stored procedure return only the Entities whose names begin with either "A" or "a".

@SystemId: The MetadataObjectId of the LobSystem, which contains the Entities to be returned. The value MUST be an [Id](#).

Return Code Values: MUST return an integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.57.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.57.2 Entity Result Set

See [Entity result set](#). The result set MUST contain zero or more rows.

3.1.5.58 proc_ar_GetEntitiesForSystemWithCount

The **proc_ar_GetEntitiesForSystemWithCount** stored procedure is invoked to get the Entities contained by the LobSystem with the specified MetadataObjectId, along with the count of such Entities.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetEntitiesForSystemWithCount (  
    @SystemId              int  
);
```

@SystemId: The MetadataObjectId of the LobSystem, which contains the Entities to be returned. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return two result sets in the following order:

3.1.5.58.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.58.2 Entity Result Set

See [Entity result set](#). The result set MUST contain zero or more rows.

3.1.5.59 proc_ar_GetEntityById

The **proc_ar_GetEntityById** stored procedure is invoked to retrieve the Entity with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetEntityById (  
    @MetadataObjectId      int  
) ;
```

@MetadataObjectId: The MetadataObjectId for the Entity that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return a single result set:

3.1.5.59.1 Entity Result Set

See [Entity Result Set](#). The result set MUST contain zero or more rows.

3.1.5.60 proc_ar_GetFilterDescriptorById

The **proc_ar_GetFilterDescriptorById** stored procedure is called to retrieve the FilterDescriptor identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetFilterDescriptorById (  
    @MetadataObjectId      int  
) ;
```

@MetadataObjectId: The MetadataObjectId of the FilterDescriptor that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.60.1 FilterDescriptor Result Set

See [FilterDescriptor result set](#). The result set MUST contain zero or one row.

3.1.5.61 proc_ar_GetFilterDescriptorsForMethodWithCount

The **proc_ar_GetFilterDescriptorsForMethodWithCount** stored procedure is called to retrieve the FilterDescriptors contained by the Method with the specified MetadataObjectId, along with the count of such FilterDescriptors.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetFilterDescriptorsForMethodWithCount (
    @MethodId          int
);
```

@MethodId: The MetadataObjectId of the Method that contains the FilterDescriptors to be returned. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets.

3.1.5.61.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.61.2 FilterDescriptor Result Set

See [FilterDescriptor result set](#). The result set MUST contain zero or more rows.

3.1.5.62 proc_ar_GetIdentifierById

The **proc_ar_GetIdentifierById** stored procedure is called to retrieve the Identifier with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetIdentifierById (
    @MetadataObjectId int
);
```

@MetadataObjectId: The MetadataObjectId of the Identifier that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.62.1 Identifier Result Set

See [Identifier result set](#). The result set MUST contain zero or more rows.

3.1.5.63 proc_ar_GetIdentifiersForEntityWithCount

The **proc_ar_GetIdentifiersForEntityWithCount** stored procedure is called to retrieve the Identifiers contained by the Entity with the specified MetadataObjectId, along with the count of such Identifiers.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetIdentifiersForEntityWithCount (
    @EntityId          int
);
```


);

@EntityId: The MetadataObjectId of the Entity that contains the Identifiers to be returned. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result sets.

3.1.5.63.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.63.2 Identifier Result Set

See [Identifier result set](#). The result set MUST contain zero or more rows.

3.1.5.64 proc_ar_GetMethodById

The **proc_ar_GetMethodById** stored procedure is called to retrieve the Method with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetMethodById (  
    @MetadataObjectId      int  
);
```

@MetadataObjectId: The MetadataObjectId of the Method that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.64.1 Method Result Set

See [Method result set](#). The result set MUST contain zero or one row.

3.1.5.65 proc_ar_GetMethodInstanceById

The **proc_ar_GetMethodInstanceById** stored procedure is called to retrieve the MethodInstance with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetMethodInstanceById (  
    @MetadataObjectId      int  
);
```

@MetadataObjectId: The MetadataObjectId of the MethodInstance that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.65.1 MethodInstance Result Set

See [MethodInstance result set](#). The result set MUST contain zero or more rows.

3.1.5.66 proc_ar_GetMethodInstancesForDataClassWithCount

The **proc_ar_GetMethodInstancesForDataClassWithCount** stored procedure is called to retrieve the MethodInstances that are contained by the DataClass with the specified MetadataObjectId, excluding those MethodInstances that are Associations, along with the count of such MethodInstances.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetMethodInstancesForDataClassWithCount (  
    @ClassId          int  
);
```

@ClassId: The MetadataObjectId of the DataClass that contains the MethodInstances to be returned. The value MUST be an [Id](#).

Return Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result sets.

3.1.5.66.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.66.2 MethodInstance Result Set

See [MethodInstance result set](#). The result set MUST contain zero or more rows.

3.1.5.67 proc_ar_GetMethodInstancesForMethodWithCount

The **proc_ar_GetMethodInstancesForMethodWithCount** stored procedure is called to retrieve the MethodInstances that are contained by the Method with the specified MetadataObjectId, excluding those MethodInstances that are Associations, along with the count of such MethodInstances.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetMethodInstancesForMethodWithCount (  
    @MethodId        int  
);
```

@MethodId: The MetadataObjectId of the Method that contains the MethodInstances to be returned. The value MUST be an [Id](#).

Return Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result sets.

3.1.5.67.1 Count Result Set

See [Count result set](#). The result set MUST contain 1 row.

3.1.5.67.2 MethodInstance Result Set

See [MethodInstance result set](#). The result set MUST contain zero or more rows.

3.1.5.68 proc_ar_GetMethodsForDataClassWithCount

The **proc_ar_GetMethodsForDataClassWithCount** stored procedure is called to retrieve the Methods contained by the DataClass with the specified MetadataObjectId, along with the count of such Methods.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetMethodsForDataClassWithCount (  
    @ClassId          int  
);
```

@ClassId: The MetadataObjectId of the DataClass that contains the Methods to be returned. The value MUST be an [Id](#).

Return Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result sets.

3.1.5.68.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.68.2 Method Result Set

See [Method result set](#). The result set MUST contain zero or more rows.

3.1.5.69 proc_ar_GetParameterById

The **proc_ar_GetParameterById** stored procedure is called to retrieve the Parameter with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetParameterById (  
    @MetadataObjectId int  
);
```

@MetadataObjectId: The MetadataObjectId of the Parameter that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.69.1 Parameter Result Set

See [Parameter Result Set](#). The result set MUST contain zero or more rows.

3.1.5.70 proc_ar_GetParametersForMethodWithCount

The **proc_ar_GetParametersForMethodWithCount** stored procedure is called to retrieve Parameters information for the Method with the given MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetParametersForMethodWithCount (
    @MethodId          int
);
```

@MethodId: The MetadataObjectId of the Method for which Parameter information will be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets.

3.1.5.70.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.70.2 Parameter Result Set

See [Parameter result set](#). The result set MUST contain zero or more rows.

3.1.5.71 proc_ar_GetPropertiesForMetadataObject

The **proc_ar_GetPropertiesForMetadataObject** stored procedure is invoked to retrieve Properties for the MetadataObject with the given MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetPropertiesForMetadataObject (
    @MetadataObjectId int,
    @ErrorCode         int OUTPUT
);
```

@MetadataObjectId: The MetadataObjectId of the MetadataObject whose Properties is to be returned. The value MUST be an [Id](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	Object not found

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.71.1 Property Result Set

The **Property** Result Set contains the name and value of the Property associated with a MetadataObject. Each row represents one Property. The result set MUST contain zero or more rows.

The T-SQL syntax for the result set is as follows:

```
Name          nvarchar(255),
Value         sql_variant;
```

Name: The programmatic name of the Property.

Value: The value of the Property.

3.1.5.72 proc_ar_GetRootTypeDescriptorForParameter

The **proc_ar_GetRootTypeDescriptorForParameter** stored procedure is called to retrieve the TypeDescriptor information that is contained by the Parameter with the specified MetadataObjectId and has no parent TypeDescriptor.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetRootTypeDescriptorForParameter (
    @ParameterId      int
);
```

@ParameterId: This parameter provides the MetadataObjectId value of an existing Parameter that contains the TypeDescriptor to be returned. The value MUST be an [Id](#).

Return Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result set.

3.1.5.72.1 TypeDescriptor Result Set

See [TypeDescriptor result set](#). The result set MUST contain zero or one row.

3.1.5.73 proc_ar_GetSystemById

The **proc_ar_GetSystemById** stored procedure is invoked to retrieve the LobSystem with the specified MetadataObjectId. The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetSystemById (
    @MetadataObjectId int
);
```

@MetadataObjectId: The MetadataObjectId of the LobSystem that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.73.1 System Result Set

See [System Result Set](#). The result set MUST contain zero or one row.

3.1.5.74 proc_ar_GetSystemDataBySystemName

The **proc_ar_GetSystemDataBySystemName** stored procedure is called to retrieve the binary an implementation-specific [<37>](#) Business Logic Module associated with the given LobSystem. The Business Logic Module can be used to provide implementation-specific [<38>](#) business logic (2) that MAY [<39>](#) be referenced by the [TypeDescriptorTypeNames](#) for the TypeDescriptors contained by the Parameters contained by the Methods contained by the Entities contained by the LobSystem specified by the value of *@SystemName*.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetSystemDataBySystemName (  
    @SystemName          nvarchar (255)  
)  
;
```

@SystemName: The programmatic name of the LobSystem. The value MUST be a [Name](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.74.1 System Data Result Set

The **System Data** Result Set contains the binary representation of the Business Logic Module associated with a single LobSystem. The result set MUST contain zero or one row.

The T-SQL syntax for the result set is as follows:

```
Length          int,  
Data            image;
```

Length: The size of the binary business logic module in bytes.

Data: The binary Business Logic Module.

3.1.5.75 proc_ar_GetSystemInstanceById

The **proc_ar_GetSystemInstanceById** stored procedure is invoked to retrieve the LobSystemInstance with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetSystemInstanceById (  
    @MetadataObjectId    int
```

);

@MetadataObjectId: The MetadataObjectId of the LobSystemInstance that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.75.1 System Instance Result Set

See [System Instance Result Set](#). The result set MUST contain zero or one row.

3.1.5.76 proc_ar_GetSystemInstancesForSystemWithCount

The **proc_ar_GetSystemInstancesForSystemWithCount** stored procedure is called to retrieve LobSystemInstances information contained by the LobSystem with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetSystemInstancesForSystemWithCount (
    @SystemId          int
);
```

@SystemId: The MetadataObjectId of the LobSystem. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Results Sets: MUST return the following result sets.

3.1.5.76.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.76.2 System Instance Result Set

See [system instance result set](#). The result set MUST contain zero or more rows.

3.1.5.77 proc_ar_GetSystemsLikeNameWithCount

The **proc_ar_GetSystemsLikeNameWithCount** stored procedure is called to retrieve the count and the LobSystems that satisfy either one of the following constraints:

- Any LobSystems that have names that match the specified pattern.
- Any LobSystems whose localized names match the specified pattern and either have the given LCID or their LCID is zero.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetSystemsLikeNameWithCount (
    @MetadataObjectName    nvarchar(255),
    @LCID                  int
);
```

);

@MetadataObjectName: The string that specifies the name pattern of the LobSystems to be returned. The characters in this string MUST be in upper case. It can include wildcard characters. For example, setting the *@MetadataObjectName* as "A%" will make this stored procedure return only the LobSystems with names beginning with either "A" or "a".

@LCID: The LCID of the localized names of the LobSystems to be fetched. In the following two cases, this parameter MUST be ignored:

- @MetadataObjectName matches the name of the LobSystem;
- @MetadataObjectName matches the localized name and the LCID of localized name is zero.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result sets.

3.1.5.77.1 Count Result Set

See [Count result set](#). The result set MUST contain one row.

3.1.5.77.2 System Result Set

See [system result set](#). The result set MUST contain zero or more rows.

3.1.5.78 proc_ar_GetTypeDescriptorById

The **proc_ar_GetTypeDescriptorById** stored procedure is called to retrieve the TypeDescriptor with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetTypeDescriptorById (  
    @MetadataObjectId          int  
);
```

@MetadataObjectId: The MetadataObjectId of the TypeDescriptor that is to be retrieved. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.78.1 TypeDescriptor Result Set

See [TypeDescriptor Result Set](#). The result set MUST contain zero or one row.

3.1.5.79 proc_ar_GetTypeDescriptorsByNameAndParameter

The **proc_ar_GetTypeDescriptorsByNameAndParameter** stored procedure is called to retrieve TypeDescriptors that have the specified name and are either the root or the child TypeDescriptor of the specified Parameter.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetTypeDescriptorsByNameAndParameter (
    @ParameterId          int,
    @Name                  nvarchar(255)
);
```

@ParameterId: The MetadataObjectId of an existing Parameter for which TypeDescriptors are to be returned. The value MUST be an [Id](#).

@Name: The name of the TypeDescriptor to be returned. The value MUST be a [Name](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following result set.

3.1.5.79.1 TypeDescriptor Result Set

See [TypeDescriptor Result Set](#). The result set MUST contain zero or more rows.

3.1.5.80 proc_ar_GetTypeDescriptorsForFilterDescriptorWithCount

The **proc_ar_GetTypeDescriptorsForFilterDescriptorWithCount** stored procedure is called to retrieve TypeDescriptors that reference the specified FilterDescriptor.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_GetTypeDescriptorsForFilterDescriptorWithCount (
    @FilterDescriptorId    int
);
```

@FilterDescriptorId: The MetadataObjectId of the FilterDescriptor for which TypeDescriptors are to be returned. The value MUST be an [Id](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST return the following two result sets in the listed order:

3.1.5.80.1 Count Result Set

See [Count Result Set](#). The result set MUST contain one row.

3.1.5.80.2 TypeDescriptor Result Set

See [TypeDescriptor Result Set](#). The result set MUST contain zero or more rows.

3.1.5.81 proc_ar_SetAccessControlEntryForMetadataObject

The **proc_ar_SetAccessControlEntryForMetadataObject** stored procedure is invoked to add an access control entry and associate it with the MetadataObject identified by the specified MetadataObjectId. If an access control entry with *@IdentityName* already exists, it is replaced by the newly created access control entry.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_SetAccessControlEntryForMetadataObject (
    @MetadataObjectId      int,
    @IdentityName          nvarchar(250),
    @DisplayName           nvarchar(250),
    @RawSid                varbinary(512),
    @Rights                 bigint
);

```

@MetadataObjectId: The MetadataObjectId of the MetadataObject to which the access control entry is to be added. The value MUST be an [Id](#).

@IdentityName: The programmatic name of the security principal (2).

@DisplayName: The name of the security principal (2) used for display purposes.

@RawSid: The SID, if the security principal (2) is a Windows security principal. If the security principal (2) is not a Windows security principal, the value MUST be NULL.

@Rights: The permissions available to the security principal (2) for the MetadataObject identifier by the MetadataObjectId. It MUST be [MetadataRights](#).

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.82 proc_ar_SetDefaultAction

The **proc_ar_SetDefaultAction** stored procedure is called to set or clear the default **Action** on the specified Entity.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_SetDefaultAction (
    @EntityId              int,
    @ActionName            nvarchar(255),
    @ErrorCode             int OUTPUT
);

```

@EntityId: The MetadataObjectId of the Entity to set or clear the DefaultAction for. This parameter MUST be non-NULL. The value MUST be an [Id](#).

@ActionName: This parameter MUST take values defined in the following table.

Value	Description
NULL	This operation clears the DefaultAction for the Entity with MetadataObjectId equal to <i>@EntityId</i> .
Not NULL	This operation sets the DefaultAction for the Entity with MetadataObjectId equal to <i>@EntityId</i> to the Action with the name <i>@ActionName</i> , if such Action is contained by this Entity; otherwise, the <i>@ErrorCode</i> parameter MUST be set to -2 and DefaultAction for this Entity MUST be unchanged.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table:

Value	Description
0	No errors encountered.
-2	The value of <i>@ActionName</i> does not match the names of any of the Actions contained by the Entity with MetadataObjectId equal to <i>@EntityId</i> .
A positive integer	A system specific error with the given error code has occurred.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<40>](#) return zero or more result sets which the protocol client MUST ignore.

3.1.5.83 `proc_ar_SetDefaultValuesForTypeDescriptor`

The `proc_ar_SetDefaultValuesForTypeDescriptor` stored procedure is called to set a DefaultValue for TypeDescriptor identified by the specified MetadataObjectId in *@TypeDescriptorId* and a MethodInstance identified by the specified MetadataObjectId in *@MethodInstanceId*.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_SetDefaultValuesForTypeDescriptor (
    @TypeDescriptorId      int,
    @MethodInstanceId      int,
    @Value                 sql_variant,
    @ErrorCode             int OUTPUT
);
```

@TypeDescriptorId: The MetadataObjectId of the TypeDescriptor associated with the DefaultValue to be set. The value MUST be an [Id](#).

@MethodInstanceId: The MetadataObjectId of the MethodInstance associated with the DefaultValue to be set. The value MUST be an Id.

@Value: The DefaultValue.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-2	At least one of the following is true: <ul style="list-style-type: none"> ▪ A TypeDescriptor with MetadataObjectId equal to <i>@TypeDescriptorId</i> is not found. ▪ A MethodInstance with MetadataObjectId equal to <i>@MethodInstanceId</i> is not found.
-3	The TypeDescriptor with MetadataObjectId equal to <i>@TypeDescriptorId</i> already has the implementation-specific maximum number of DefaultValues allowed.
-600	Parameter of the TypeDescriptor with MetadataObjectId equal to <i>@TypeDescriptorId</i> is not contained by the Method that contains MethodInstance with MetadataObjectId equal to <i>@MethodInstanceId</i> .

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.84 **proc_ar_SetSystemDataBySystemName**

The **proc_ar_SetSystemDataBySystemName** stored procedure is called to store the binary an implementation-specific [<41>](#) Business Logic Module associated with the given LobSystem. The Business Logic Module can be used to provide implementation-specific [<42>](#) business logic (2) that MAY [<43>](#) be referenced by the [TypeDescriptorTypeNames](#) for the TypeDescriptors contained by the Parameters contained by the Methods contained by the Entities contained by the LobSystem specified by the value of *@SystemName*.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_SetSystemDataBySystemName (  
    @SystemName          nvarchar(255),  
    @AssemblyName       nvarchar(255),  
    @Length              int,  
    @Data                image  
);
```

@SystemName: The programmatic name of the LobSystem to set the Business Logic Module for. The value of this parameter MUST match the programmatic name of an existing LobSystem in the metadata store. The value MUST be a [Name](#).

@AssemblyName: The **Business Logic Module Reference**.

@Length: The size of the binary Business Logic Module in bytes.

@Data: The binary Business Logic Module.

Return Values: An integer that the protocol client MUST ignore.

Result Sets: MUST NOT return any result sets.

3.1.5.85 **proc_ar_UpdateActionById**

The **proc_ar_UpdateActionById** stored procedure is called to change the attributes of the Action identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_UpdateActionById (  
    @Id                  int,  
    @Name                nvarchar(50),  
    @IsCached            bit,  
    @Version              int OUTPUT,  
    @Position            int,  
    @IsDisplayed          bit,  
    @IsOpenedInNewWindow bit,  
    @Icon                nvarchar(2080),  
    @URL                 nvarchar(2080),  
    @ErrorCode           int OUTPUT  
);
```

@Id: The MetadataObjectId of the Action that is to be updated. The value MUST be an [Id](#).

@Name: The name of the Action. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Action is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time Action with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the Action is updated. The value MUST wrap around after reaching 2147483646.

@Position: The order of this Action among the other Actions displayed in a user interface for this Entity. The value MUST be a [Position](#).

@IsDisplayed: A bit that provides a hint on whether the Action is displayed in the user interface presented to the user. The value MUST be an [IsDisplayed](#).

@IsOpenedInNewWindow: A bit that a hint on whether the results of executing the Action are displayed in a new window in the user interface presented to the user. The value MUST be an [IsOpenedInNewWindow](#).

@Icon: The URL of the icon associated with this Action. The value MUST be an [Icon](#).

@URL: The URL associated with this Action. The value MUST be an [URL](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Entity that contains this Action already contains another Action with the specified <i>@Name</i> .
-2	An Action with the specified <i>@Id</i> does not exist.
-6	An Action with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Action.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<44>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.86 **proc_ar_UpdateActionParameterById**

The **proc_ar_UpdateActionParameterById** stored procedure is called to change the attributes of the ActionParameter identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_UpdateActionParameterById (  
    @Id                int,  
    @IsCached          bit,
```

```

@Version          int OUTPUT,
@Name             nvarchar(50),
@Index           tinyint,
@ErrorCode       int OUTPUT
);

```

@Id: The MetadataObjectId of the ActionParameter that is to be updated. The value MUST be an [Id](#).

@IsCached: A bit that specifies whether this ActionParameter is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time ActionParameter with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the ActionParameter is updated. The value MUST wrap around after reaching 2147483646.

@Name: The name of the ActionParameter. The value MUST be a [Name](#).

@Index: A value, indicating the position of this ActionParameter among the ActionParameters of the Action that contains this ActionParameter. It MUST be [Index](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Action that contains this ActionParameter already contains another ActionParameter with the specified <i>@Name</i> .
-2	An ActionParameter with the specified <i>@Id</i> does not exist.
-6	An ActionParameter with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the ActionParameter.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<45>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.87 `proc_ar_UpdateAssociationById`

The `proc_ar_UpdateAssociationById` stored procedure is called to change the attributes of the Association identified by its given MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateAssociationById (
@Id             int,
@Name          nvarchar(255),
@IsCached      bit,
@ReturnTypeDescriptorId int,
@Type         tinyint,
@Version       int OUTPUT,

```

```

        @ErrorCode          int OUTPUT
    );

```

@Id: The MetadataObjectId of the Association that is to be updated. The value MUST be an [Id](#).

@Name: The name of the Association. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Association is frequently used. The value MUST be an [IsCached](#).

@ReturnTypeDescriptorId: The MetadataObjectId of the ReturnTypeDescriptor. The value MUST be an Id. It MUST be equal to the ReturnTypeDescriptor specified when the Association was created.

@Type: The type of the Association. This MUST be 4.

@Version: The value of version at the time Association with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the Association is updated. The value MUST wrap around after reaching 2147483646.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The LobSystem already has another Association with MetadataObject name equal to <i>@Name</i> .
-2	An Association with MetadataObjectId equal to <i>@Id</i> is not available.
-6	Association with MetadataObjectId equal to <i>@Id</i> has been updated by a context other than the one that it has been currently read by. This happens when the value of <i>@Version</i> does not match with the version for the Association.
-500	This happens when the value of <i>@ReturnTypeDescriptorId</i> does not match with the MetadataObjectId of the ReturnTypeDescriptor of the Association or if the value of <i>@Type</i> does not match with the MethodInstanceType for the Association, which is set to 4 on creation.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<46>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.88 **proc_ar_UpdateEntityById**

The **proc_ar_UpdateEntityById** stored procedure is invoked to change the attributes of the Entity identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateEntityById (
    @Id          int,
    @Name        nvarchar(255),
    @IsCached    bit,

```

```

@Version          int OUTPUT,
@SystemId         int,
@EstimatedInstanceCount int,
@ErrorCode        int OUTPUT
);

```

@Id: The identifier for the Entity that is to be updated. The value MUST be an [Id](#).

@Name: The name of the Entity. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Entity is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time Entity with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the Entity is updated. The value MUST wrap around after reaching 2147483646.

@SystemId: The MetadataObjectId of the LobSystem with which the Entity is associated. This MUST be an Id. This MUST be the MetadataObjectId of a LobSystem currently in the metadata store.

@EstimatedInstanceCount: The estimated number of instances of this Entity present within the LobSystemInstance. The value MUST be an [EstimatedInstanceCount](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The LobSystem that contains this Entity already contains another Entity with <i>@Name</i> .
-2	An Entity with the specified <i>@Id</i> does not exist.
-3	The LobSystem already contains the implementation-specific maximum allowed number of Entities.
-6	An Entity with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Entity.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<47>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.89 `proc_ar_UpdateFilterDescriptorById`

The `proc_ar_UpdateFilterDescriptorById` stored procedure is invoked to change the attributes of the FilterDescriptor identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateFilterDescriptorById (
    @Id          int,

```



```

    @Name                nvarchar(255),
    @IsCached            bit,
    @Version              int OUTPUT,
    @TypeName            nvarchar(255),
    @ErrorCode           int OUTPUT
);

```

@Id: The MetadataObjectId of the FilterDescriptor that is to be updated. The value MUST be an [Id](#).

@Name: The name of the FilterDescriptor. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this FilterDescriptor is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time FilterDescriptor with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the FilterDescriptor is updated. The value MUST wrap around after reaching 2147483646.

@TypeName: The type name of the FilterDescriptor. The value MUST be an [FilterDescriptorTypeName](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Method that contains this FilterDescriptor already contains another FilterDescriptor with the specified <i>@Name</i> .
-2	A FilterDescriptor with the specified <i>@Id</i> does not exist.
-6	A FilterDescriptor with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the FilterDescriptor.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<48>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.90 `proc_ar_UpdateIdentifierById`

The `proc_ar_UpdateIdentifierById` stored procedure is invoked to change the attributes of the Identifier identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateIdentifierById (
    @Id                int,
    @Name              nvarchar(255),
    @IsCached          bit,
    @Version            int OUTPUT,
    @TypeName          nvarchar(255),

```

```

        @ErrorCode          int OUTPUT
    );

```

@Id: The MetadataObjectId of the Identifier that is to be updated. The value MUST be an [Id](#).

@Name: The name of the Identifier. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Identifier is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time Identifier with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the Identifier is updated. The value MUST wrap around after reaching 2147483646.

@TypeName: The type name of the Identifier. The value MUST be an [IdentifierTypeName](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Entity that contains this Identifier already contains another Identifier with the specified <i>@Name</i> .
-2	An Identifier with the specified <i>@Id</i> does not exist.
-6	An Identifier with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Identifier.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<49>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.91 `proc_ar_UpdateMethodById`

The `proc_ar_UpdateMethodById` stored procedure is invoked to change the attributes of the Method identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateMethodById (
    @Id                int,
    @Name              nvarchar(255),
    @IsCached          bit,
    @Version            int OUTPUT,
    @IsStatic          bit,
    @ErrorCode         int OUTPUT
);

```

@Id: The MetadataObjectId of the Method that is to be updated. The value MUST be an [Id](#).

@Name: The name of the Method. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this Method is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time Method with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the Method is updated. The value MUST wrap around after reaching 2147483646.

@IsStatic: A bit specifying whether the Method is associated with an EntityInstance. The value MUST be an [IsStatic](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The Entity that contains this Method already contains another Method with the specified <i>@Name</i> .
-2	A Method with the specified <i>@Id</i> does not exist.
-6	A Method with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the Method.
A positive integer	A T-SQL error code..

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<50>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.92 **proc_ar_UpdateMethodInstanceById**

The **proc_ar_UpdateMethodInstanceById** stored procedure is called to change attributes of MethodInstance with given MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_UpdateMethodInstanceById (  
    @Id                int,  
    @Name              nvarchar(255),  
    @IsCached          bit,  
    @Version           int OUTPUT,  
    @ReturnTypeDescriptorId int,  
    @Type              tinyint,  
    @ErrorCode         int OUTPUT  
);
```

@Id: This parameter MUST contain MetadataObjectId of the MethodInstance to be updated. The value MUST be an [Id](#).

@Name: This parameter MUST be used to set the value of programmatic name of MethodInstance. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this MethodInstance is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time MethodInstance with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the MethodInstance is updated. The value MUST wrap around after reaching 2147483646.

@ReturnTypeDescriptorId: The MetadataObjectId of the ReturnTypeDescriptor. The TypeDescriptor MUST exist in the metadata store. The value MUST be an Id.

@Type: The type of the MethodInstance. The value MUST be an [MethodInstanceType](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-1	The Method that contains this MethodInstance already contains another MethodInstance with MetadataObject name equal to <i>@Name</i> .
-2	A MethodInstance with MetadataObjectId equal to <i>@Id</i> doesn't exist.
-6	The MethodInstance with MetadataObjectId equal to <i>@Id</i> has been updated by a context other than the one that it has been currently read by. This happens when the value of <i>@Version</i> does not match with the version for the MethodInstance.
-200	An Entity cannot contain more than one Method that contains at most one MethodInstance of MethodInstanceType Finder, or a Method that contains more than one MethodInstance with MethodInstanceType Finder.
-201	An Entity cannot contain more than one Method that contains at most one MethodInstance with MethodInstanceType SpecificFinder, or a Method that contains more than one MethodInstance with MethodInstanceType SpecificFinder.
-202	An Entity cannot contain more than one Method that contains at most one MethodInstance with MethodInstanceType IdEnumerator, or a Method that contains more than one MethodInstance with MethodInstanceType IdEnumerator.
-203	Method that contains Parameter that contains TypeDescriptor with MetadataObjectId equal to <i>@ReturnTypeDescriptorId</i> , does not contain the MethodInstance with MetadataObjectId equal to <i>@Id</i> .
-204	Parameter that contains TypeDescriptor with MetadataObjectId equal to <i>@ReturnTypeDescriptor</i> is a Parameter with Direction of 1.
-205	An Entity cannot contain more than one Method that contains at most one MethodInstance with MethodInstanceType AccessChecker, or a Method that contains more than one MethodInstance with MethodInstanceType AccessChecker.
A positive integer	A T-SQL error code.

Return Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<51>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.93 proc_ar_UpdateParameterById

The **proc_ar_UpdateParameterById** stored procedure is called to change the attributes of the Parameter identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```
PROCEDURE proc_ar_UpdateParameterById (
    @Id                int,
    @Name              nvarchar(255),
    @IsCached          bit,
    @Version            int OUTPUT,
    @OrdinalNumber     tinyint OUTPUT,
    @Direction         tinyint,
    @TypeReflectorTypeName nvarchar(255),
    @ErrorCode         int OUTPUT
);
```

@Id: This parameter MUST contain MetadataObjectId of the Parameter to be updated.

@Name: This parameter MUST be used to set the value of programmatic name of the Parameter.

@IsCached: A bit that specifies whether this Parameter is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time Parameter with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the Parameter is updated. The value MUST wrap around after reaching 2147483646.

@OrdinalNumber: The position of the Parameter in the Parameter signature of the Method containing this Parameter. If the position is the same as another Parameter's position for the same parent Method, the other Parameter's position, along with all Parameters positioned subsequently are incremented. When the stored procedure returns, all Parameters of the Method containing this Parameter MUST have positions in the range 0 to X, where X+1 is the number of Parameters in the Method. Parameters in the Method other than this Parameter MUST NOT have their relative positioning altered.

@Direction: This parameter MUST be used to set the direction in which the Parameter is passed. It MUST be [Direction](#).

@TypeReflectorTypeName: The type name of the TypeReflector that will be used to resolve the native type of this parameter. The value MUST be a [TypeReflectorTypeName](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-1	The Method that contains this Parameter already contains another Parameter with MetadataObject name equal to <i>@Name</i> .
-2	A Parameter with MetadataObjectId equal to <i>@Id</i> doesn't exist.
-6	Parameter with MetadataObjectId equal to <i>@Id</i> has been updated by a context other than the one that it has been currently read by. This happens when the value of <i>@Version</i> does not match with

Value	Description
	the Version for the Parameter.
-100	The Method that contains this Parameter already contains another Parameter with Direction of 4.
-102	The Parameter with MetadataObjectId equal to <i>@Id</i> cannot be updated to have Direction of 1. There is a MethodInstance with ReturnPropertyDescriptor whose MetadataObjectId is equal to MetadataObjectId of a TypeDescriptor in the TypeDescriptor tree of root TypeDescriptor of this Parameter.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<52>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.94 **proc_ar_UpdateSystemById**

The **proc_ar_UpdateSystemById** stored procedure is called to change the attributes of the LobSystem identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateSystemById (
    @Id                int,
    @Name              nvarchar(255),
    @IsCached          bit,
    @Version            int OUTPUT,
    @SystemUtilityTypeName nvarchar(255),
    @ConnectionManagerTypeName nvarchar(255),
    @EntityInstanceTypeName nvarchar(255),
    @ErrorCode          int OUTPUT
);

```

@Id: The MetadataObjectId of the LobSystem that will be updated. The value MUST be an [Id](#).

@Name: The name of the LobSystem. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this LobSystem is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time LobSystem with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the LobSystem is updated. The value MUST wrap around after reaching 2147483646.

@SystemUtilityTypeName: The name of the System Utility that will be used to execute the Methods in this LobSystem. The value MUST be a [SystemUtilityTypeName](#).

@ConnectionManagerTypeName: The name of the connection manager that will be used while connecting to this LobSystem. The value MUST be a [ConnectionManagerTypeName](#).

@EntityInstanceTypeName: The name of the unit of implementation-specific [<53>](#) business logic (2) that will be used to create the objects that will carry EntityInstance data to client applications. The value MUST be a [EntityInstanceTypeName](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	A LobSystem with the specified <i>@Name</i> already exists in the metadata store.
-2	A LobSystem with the specified <i>@Id</i> does not exist.
-6	A LobSystem with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the LobSystem.
Positive Integer Number	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<54>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.5.95 `proc_ar_UpdateSystemInstanceById`

The `proc_ar_UpdateSystemInstanceById` stored procedure is called to change the attributes of the LobSystemInstance identified by the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateSystemInstanceById (
    @Id                int,
    @Name              nvarchar(255),
    @IsCached          bit,
    @Version           int OUTPUT,
    @SystemId          int,
    @ErrorCode         int OUTPUT
);

```

@Id: The MetadataObjectId of the LobSystemInstance that will be updated. The value MUST be an [Id](#).

@Name: The name of the LobSystemInstance. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this LobSystemInstance is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time LobSystemInstance with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the LobSystemInstance is updated. The value MUST wrap around after reaching 2147483646.

@SystemId: The MetadataObjectId of the LobSystem that contains this LobSystemInstance. The value MUST be a LobSystem that currently exists in the metadata store.

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer listed in the following table.

Value	Description
0	No errors encountered.
-1	The LobSystemInstance already contains another Entity with the specified @Name.
-2	A LobSystemInstance with the specified @Id is not available.
-3	The LobSystem with @SystemId already contains implementation-specific maximum number of LobSystemInstances.
-6	A LobSystemInstance with the specified MetadataObjectId has been updated by a context other than the one that it has been currently read by. This happens when the version specified does not match with the current version of the LobSystemInstance.
A positive integer	A T-SQL error code.

Return Code Values: An integer that the protocol client MUST ignore.

Result Sets: MAY <55> return zero or more result sets that the protocol client MUST ignore.

3.1.5.96 proc_ar_UpdateTypeDescriptorById

The **proc_ar_UpdateTypeDescriptorById** stored procedure is called to change attributes of TypeDescriptor with the specified MetadataObjectId.

The T-SQL syntax for the stored procedure is as follows:

```

PROCEDURE proc_ar_UpdateTypeDescriptorById (
    @Id                int,
    @Name              nvarchar(255),
    @IsCached          bit,
    @Version           int OUTPUT,
    @ParentTypeDescriptorId int,
    @TypeName          nvarchar(255),
    @IdentifierId      int,
    @FilterDescriptorId int,
    @IsCollection      bit,
    @ErrorCode         int OUTPUT,
    @ContainsIdentifier bit OUTPUT,
    @ContainsFilterDescriptor bit OUTPUT
);

```

@Id: The MetadataObjectId of the TypeDescriptor to be updated. The value MUST be an [Id](#).

@Name: The programmatic name of TypeDescriptor to be updated. The value MUST be a [Name](#).

@IsCached: A bit that specifies whether this TypeDescriptor is frequently used. The value MUST be an [IsCached](#).

@Version: The value of version at the time TypeDescriptor with the specified MetadataObjectId was last read. This value MUST be incremented in the metadata store every time the TypeDescriptor is updated. The value MUST wrap around after reaching 2147483646.

@ParentTypeDescriptorId: The MetadataObjectId of the parent **TypeDescriptor** that contains this TypeDescriptor. If not NULL, The value MUST be a TypeDescriptor that currently exists in the metadata store. The value MUST be an Id.

@TypeName: The programmatic name of the data type that is represented by this TypeDescriptor. The value MUST be a [TypeDescriptorTypeName](#).

@IdentifierId: The MetadataObjectId of the **Identifier** referenced by this TypeDescriptor. The value MUST be an Id.

@FilterDescriptorId: The MetadataObjectId of the **FilterDescriptor** associated with this TypeDescriptor. The value MUST be an Id.

@IsCollection: A bit that specifies whether this TypeDescriptor is to be interpreted by protocol clients as a collection of native LOB System data structures. The value MUST be an [IsCollection](#).

@ErrorCode: The error code. Upon return from this stored procedure, this parameter MUST be set to an integer that is listed in the following table.

Value	Description
0	No errors encountered.
-1	The TypeDescriptor that contains this TypeDescriptor already contains another TypeDescriptor with MetadataObject name equal to <i>@Name</i> .
-2	A TypeDescriptor with MetadataObjectId equal to <i>@Id</i> doesn't exist.
-3	At least one of the following has happened: <ul style="list-style-type: none"> ▪ The TypeDescriptor with MetadataObjectId equal to <i>@ParentTypeDescriptorId</i> already contains implementation-specific maximum number of TypeDescriptors allowed. ▪ The FilterDescriptor with MetadataObjectId equal to <i>@FilterDescriptorId</i> already associated with implementation-specific maximum number of TypeDescriptors allowed.
-6	TypeDescriptor with MetadataObjectId equal to <i>@Id</i> has been updated by a context other than the one that it has been currently read by. This happens when the value of <i>@Version</i> does not match with the version for the TypeDescriptor.
-300	The Parameter of the TypeDescriptor with MetadataObjectId equal to <i>@Id</i> already has a TypeDescriptor hierarchy deeper than the implementation-specific maximum level allowed.
-302	The Parameter of the TypeDescriptor with MetadataObjectId equal to <i>@Id</i> already has a root TypeDescriptor.
-303	The Method that contains FilterDescriptor with MetadataObjectId equal to <i>@FilterDescriptorId</i> does not contain the Parameter of the TypeDescriptor with MetadataObjectId equal to <i>@Id</i> .
-304	The TypeDescriptor tree of root TypeDescriptor of Parameter that contains TypeDescriptor with MetadataObjectId equal to <i>@Id</i> does not contain the TypeDescriptor with MetadataObjectId equal to <i>@ParentTypeDescriptorId</i> .
-305	A TypeDescriptor with <i>IsCollection</i> attribute set to true cannot contain another TypeDescriptor with <i>IsCollection</i> attribute set to true.
-306	A TypeDescriptor with <i>IsCollection</i> attribute set to true cannot contain more than one TypeDescriptor.

Value	Description
A positive integer	A T-SQL error code.

@ContainsIdentifier: A Boolean value specifying if any TypeDescriptor in the TypeDescriptor tree of this TypeDescriptor references an Identifier.

@ContainsFilterDescriptor: A Boolean value specifying if any TypeDescriptor in the TypeDescriptor tree of this TypeDescriptor has an associated FilterDescriptor.

Return Values: An integer that the protocol client MUST ignore.

Result Sets: MAY [<56>](#) return zero or more result sets that the protocol client MUST ignore.

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Metadata Client Details

The **Metadata Client** acts as a client when it calls the **back-end database server** requesting processing of stored procedures and optionally caching some of the data retrieved by the stored procedures.

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The MetadataObjects stored in the metadata store may be maintained as object structures within the protocol client.

The protocol client sends messages to the protocol server to add, retrieve, change, and delete MetadataObjects stored in the protocol server.

3.2.1.1 MetadataObject Caching

The **Metadata Client** can cache the following sets of data for this protocol within object structures. Data within these structures may not be a complete representation of all data on the **back-end database server**, but can be populated as various requests to the **back-end database server** are fulfilled. Data may be cached at two levels independently - the **MetadataObjects** themselves as well as the relationships between MetadataObjects of different types.

Data maintained in the Metadata Client can be discarded after individual sequences of requests have completed as part of the cache invalidation mechanism. Cache invalidation can happen independently for objects and relationships. To trigger cache invalidation, the protocol client MUST

call [proc ar BumpCacheInvalidationCounter](#) with the type of the MetadataObject whose cache is to be invalidated along with the type (Object or Relationship) cache to be invalidated.

Note that the preceding conceptual data can be implemented using a variety of techniques. An implementation is at liberty to implement such data in any way it pleases.

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

The Protocol Client handles each stored procedure with the same basic processing method of calling the stored procedure and waiting for the result code and any result sets that will be returned.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Examples

This section provides specific example scenarios for operations on stored MetadataObjects. These examples describe in detail the process of communication between the protocol server and protocol client. In conjunction with the detailed client and server protocol specification in section 3, this information is intended to provide a comprehensive view on how the protocol client operates with the protocol server when executing such an operation.

The examples manipulate Entities. However, the principals illustrated apply equally to other MetadataObjects.

4.1 Creating an Entity

This example illustrates how a user can create an Entity in the metadata store.

The example assumes that:

- A LobSystem that will contain the Entity is already created in the metadata store.
- The LobSystem is identified by a MetadataObjectId of 33.
- The LobSystem has two access control entries associated with it:
 - The first authorizes 'domain\user1' with [MetadataRights](#) 'Edit', 'Execute'.
 - The second authorizes 'domain\user2' with MetadataRights 'Execute'

The following actions are carried out:

1. The user requests the protocol client to create an Entity with name 'Customer' and estimated instance count of '100'.
2. The protocol client calls the [proc_ar_CreateEntity](#) stored procedure:

```
exec @return_value = proc_ar_CreateEntity
    @Name = 'Customer',
    @IsCached = 1,
    @SystemId = 33,
    @EstimatedInstanceCount = 100,
    @CreatedId OUTPUT,
    @ErrorCode OUTPUT
```

3. The protocol server creates the Entity in the metadata store. It also copies the access control entries of the LobSystem and associates them with the newly created Entity. Finally, it sets `@ErrorCode` to 0.
4. The protocol server returns a variable number of result sets that the protocol client ignores.
5. The protocol server returns a return code that the protocol client ignores.
6. The protocol client returns the `@createdId` and `@errorCode` values to the user.
7. The user inspects the `@errorCode` to see if the creation was successful.
8. The user saves the `@createdId` as the MetadataObjectId of the newly created Entity for subsequent use. Assume the value of `@createdId` is 34.

4.2 Reading the Security Information of a MetadataObject

This example shows how a user can read the access control entries of an Entity.

The example assumes that the preceding example has been successfully executed.

The following actions are carried out:

1. The user requests the protocol client to read access control entries for the Entity identified by MetadataObjectId 34.
2. The protocol client calls the [proc_ar_GetAccessControlEntriesForMetadataObject](#) stored procedure.

```
exec @return_value = proc_ar_GetAccessControlEntriesForMetadataObject
    @MetadataObjectId = 34,
    @ErrorCode [int] OUTPUT
```

3. The protocol server checks whether a MetadataObject with MetadataObjectId 34 exists in the metadata store.
4. The protocol server retrieves the attributes of each of the two access control entries associated with Entity that were created in the previous example.
5. The protocol server returns a [Access Control Entry Result Set](#) with two rows to the protocol client.
6. The protocol server returns a return code that the protocol client ignores.
7. The user utilizes the access control entry information to make an implementation-specific authorization decision.

4.3 Reading an Entity

This example shows how a user can read an Entity in the metadata store.

The example assumes that the preceding example has been successfully executed.

The following actions are carried out:

1. The user requests the protocol client to read Entity with MetadataObjectId equal to 34.
2. The protocol client calls the [proc_ar_GetEntityById](#) stored procedure.

```
exec @return_value = proc_ar_ReadEntityById
    @MetadataObjectId = 34
```

3. The protocol server checks whether an Entity with MetadataObjectId 34 exists in the metadata store.
4. If it exists, the protocol server retrieves the attributes of the stored Entity.
5. The protocol server returns an [Entity Result Set](#) with one row to the protocol client. The columns in the row and the values are as follows:

```
Id      34
```

```
EstimatedInstanceCount    100
SystemId                   33
Name                       Customer
IsCached                   1
Version                    0
```

6. The protocol server returns a return code that the protocol client ignores.
7. The user retrieves the Entity attributes from the result set.

4.4 Updating an Entity

This example shows how a user can update an Entity in the metadata store.

The example assumes that the preceding example has been successfully executed.

The following actions are carried out:

1. The user requests the protocol client to update Entity with MetadataObjectId equal to 34 and change its name from "Customer" to "Buyer".
2. The protocol client calls the [proc_ar_UpdateEntityById](#) stored procedure. Attributes other than 'name' are supplied with the values obtained when the Entity was read in the preceding example.

```
exec @return_value = proc_ar_UpdateEntityById
    @Id = 34,
    @Name = 'Buyer',
    @IsCached = 1,
    @Version = 0 OUTPUT,
    @SystemId = 33,
    @EstimatedInstanceCount = 10,
    @ErrorCode OUTPUT
```

3. The protocol server checks whether an Entity with MetadataObjectId 34 exists in the metadata store.
4. If it exists, the protocol server compares the value of *@Version* with the value of the stored version for the Entity with MetadataObjectId 34. Because they are same, the protocol server updates the all the attribute of the Entity with the supplied values, increments the version counter from 0 to 1 and sets the *@ErrorCode* to 0.
5. The protocol server returns a variable number of result sets that the protocol client ignores.
6. The protocol server returns a return code that the protocol client ignores.
7. The protocol client returns the *@errorCode* and *@version* values to the user.
8. The user inspects the *@errorCode* to see if the update was successful. The user saves the *@version* value, whose value is 1, for use in subsequent updates to the Entity.

4.5 Deleting an Entity

This example shows how a user can delete an Entity in the metadata store.

The example assumes that the preceding example has been successfully executed.

The following actions are carried out:

1. The user requests the protocol client to delete Entity with MetadataObjectId equal to 34.
2. The protocol client calls the [proc_ar_DeleteEntityById](#) stored procedure.

```
exec @return_value = proc_ar_DeleteEntityById
    @Id = 34,
    @Version = 1,
    @ErrorCode OUTPUT
```

3. The protocol server checks whether an Entity with MetadataObjectId 34 exists in the metadata store.
4. If it exists, the protocol server compares the value of *@Version* with the value of the stored version for the Entity with MetadataObjectId 34. Because they are same, the protocol server deletes the Entity along with the associated Properties, localized names and access control entries and sets *@ErrorCode* to 0.
5. The protocol server returns a variable number of result sets that the protocol client ignores.
6. The protocol server returns a return code that the protocol client ignores
7. The protocol client returns the *@errorCode* values to the user
8. The user inspects the *@errorCode* to see if the deletion was successful

4.6 Cache Invalidation

This example shows how a user can invalidate cached metadata objects and relationships after one or more MetadataObjects have been created, updated or deleted.

The example assumes that the preceding example has been successfully executed. The user wants the Entity named 'Buyer', that is currently reflected in any in-memory cached metadata representations that may be maintained by a protocol client, but has been deleted from the metadata store, to also be removed from the in-memory representations.

The following actions are carried out:

1. The user requests the protocol client to remove all cached Entities from memory.
2. The protocol client calls the [proc_ar_BumpCacheInvalidationCounter](#) stored procedure.

```
exec @return_value = proc_ar_BumpCacheInvalidationCounter
    @MetadataObjectType =
        'Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Entity',
    @ObjectCache = 1
```

3. The protocol server increments the object cache version stamp for the Entity [MetadataObjectType](#).
4. The protocol server returns a return code that the protocol client ignores
5. The user requests the protocol client to remove all references to all Entities that are held by all cached MetadataObjects.

6. The protocol client calls the `proc_ar_BumpCacheInvalidationCounter` stored procedure.

```
exec @return_value = proc_ar_BumpCacheInvalidationCounter
    @MetadataObjectType =
        'Microsoft.Office.Server.ApplicationRegistry.MetadataModel.Entity',
    @ObjectCache = 0
```

7. The protocol server increments the relationship cache version stamp for the Entity `MetadataObjectType`.

8. The protocol server returns a return code that the protocol client ignores.

In parallel to the preceding process, a cache invalidation timer job is polling the cache version stamp values in the metadata store periodically. When the timer is signaled, the following actions are carried out:

1. The protocol client timer event handler calls the `proc_ar_GetCacheInvalidationCountersWithCount` stored procedure.

```
exec @return_value = proc_ar_GetCacheInvalidationCounters
```

2. The protocol server retrieves the cache version stamp values for all `MetadataObjectTypes` along with how many types there are counters for.

The protocol server returns an [Count Result Set](#) with one row to the protocol client.

3. The protocol server returns a Cache Version Stamps Result Set with as many rows as were indicated in the subsequent step to the protocol client.

4. The protocol server returns a return code that the protocol client ignores.

5. The protocol client compares the returned counter values with the values it read when the timer was previously signaled, and finds that the Object Cache Version Stamp and the Relationship Cache version stamp values are different. In response, the protocol client deletes the cached Entity references and the cached Entity `MetadataObjects` from memory.

5 Security

5.1 Security Considerations for Implementers

There are no additional security considerations for implementers. Security assumptions of this protocol are documented in [Prerequisites/Preconditions](#).

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® Office SharePoint® Server 2007
- Microsoft® SharePoint® Server 2010
- Microsoft® SQL Server® 2005
- Microsoft® SQL Server® 2008
- Microsoft® SQL Server® 2008 R2

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

[<1> Section 2.2.2.18](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

[<2> Section 2.2.2.18](#): A Business Logic Module that conforms to the [\[ECMA-335\]](#) specification and is understood by the .NET Framework.

[<3> Section 2.2.5.13](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

[<4> Section 2.2.5.14](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

[<5> Section 2.2.5.14](#): A business logic (2) module that conforms to the [\[ECMA-335\]](#) specification and is understood by the .NET Framework.

[<6> Section 3.1.1](#): Office SharePoint Server 2007 can only navigate a relationship.

[<7> Section 3.1.5.6](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<8> Section 3.1.5.7](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<9> Section 3.1.5.8](#): The current implementation does not verify this restriction. Instead, it verifies that the Entity with MetadataObjectId equal to @DestinationEntityId serves as a source for less than 1000 Associations.

[<10> Section 3.1.5.8](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<11> Section 3.1.5.9](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<12> Section 3.1.5.10](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<13> Section 3.1.5.11](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<14> Section 3.1.5.12](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<15> Section 3.1.5.13](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<16> Section 3.1.5.14](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<17> Section 3.1.5.15](#): A database that is stored on a back-end database server and contains all stored procedures and storage for the MetadataObject types.

[<18> Section 3.1.5.15](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

[<19> Section 3.1.5.15](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<20> Section 3.1.5.16](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<21> Section 3.1.5.17](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<22> Section 3.1.5.18](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<23> Section 3.1.5.19](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<24> Section 3.1.5.20](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

[<25> Section 3.1.5.21](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<26> [Section 3.1.5.22](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<27> [Section 3.1.5.23](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<28> [Section 3.1.5.24](#): The current implementation does not update Version numbers for the remaining Identifiers that are contained by the same Entity which contained the deleted Identifier.

<29> [Section 3.1.5.24](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<30> [Section 3.1.5.27](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<31> [Section 3.1.5.28](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<32> [Section 3.1.5.29](#): The current implementation does not update Version numbers for the remaining Parameters that are contained by the same Method which contained the deleted Parameter.

<33> [Section 3.1.5.29](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<34> [Section 3.1.5.32](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<35> [Section 3.1.5.33](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<36> [Section 3.1.5.34](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<37> [Section 3.1.5.74](#): A Business Logic Module that conforms to the [\[ECMA-335\]](#) specification and is understood by the .NET Framework.

<38> [Section 3.1.5.74](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

<39> [Section 3.1.5.74](#): The current implementation uses this Business Logic Module only for LobSystems that are physically represented by Web services.

<40> [Section 3.1.5.82](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<41> [Section 3.1.5.84](#): A business logic (2) module that conforms to the [\[ECMA-335\]](#) specification and is understood by the .NET Framework.

<42> [Section 3.1.5.84](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

<43> [Section 3.1.5.84](#): The current implementation uses this business logic (2) module only for LobSystems that are physically represented by Web services.

<44> [Section 3.1.5.85](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<45> [Section 3.1.5.86](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<46> [Section 3.1.5.87](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<47> [Section 3.1.5.88](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<48> [Section 3.1.5.89](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<49> [Section 3.1.5.90](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<50> [Section 3.1.5.91](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<51> [Section 3.1.5.92](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<52> [Section 3.1.5.93](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<53> [Section 3.1.5.94](#): A data type providing business logic (2) that conforms to the [\[ECMA-335\]](#) specification and can be executed by the .NET Framework.

<54> [Section 3.1.5.94](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<55> [Section 3.1.5.95](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

<56> [Section 3.1.5.96](#): Zero or more result sets are returned that are concerned with implementation-specific locking and data integrity validation functionality that the protocol client MUST ignore.

7 Change Tracking

This section identifies changes that were made to the [MS-BDCSP] protocol document between the November 2010 and December 2010 releases. 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.
- An extensive rewrite, addition, or deletion of major portions of content.
- Changes made for template compliance.
- 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 language and 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 or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

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.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

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

Some important terms used in revision 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 protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change Type
6 Appendix A: Product Behavior	Updated the list of applicable product versions.	N	Content updated.

8 Index

A

Abstract data model
[client](#) 98
[MetadataObject caching](#) 98
[server](#) 28
[Access Control Entry result set](#) 58
Action result set ([section 2.2.5.1](#) 19, [section 3.1.5.37.1](#) 59, [section 3.1.5.40.2](#) 60)
ActionParameter result set ([section 2.2.5.2](#) 20, [section 3.1.5.38.1](#) 59, [section 3.1.5.39.2](#) 59)
[Applicability](#) 11
Association result set ([section 2.2.5.3](#) 20, [section 3.1.5.45.1](#) 63, [section 3.1.5.46.1](#) 63, [section 3.1.5.47.2](#) 64, [section 3.1.5.48.2](#) 64, [section 3.1.5.49.2](#) 65)

B

[Binary structures - overview](#) 19
[Bit fields - overview](#) 19

C

[Cache invalidation example](#) 103
[Cache Version Stamps result set](#) 65
[Capability negotiation](#) 11
[Change tracking](#) 111
Client
[abstract data model](#) 98
[higher-layer triggered events](#) 99
[initialization](#) 99
[local events](#) 99
[message processing](#) 99
[Metadata Client details interface](#) 98
[MetadataObject caching](#) 98
[overview](#) 98
[sequencing rules](#) 99
[timer events](#) 99
[timers](#) 99
Common data types
[overview](#) 12
Common fields
[ConnectionManagerTypeName](#) 17
[Direction](#) 17
[EstimatedInstanceCount](#) 12
[FilterDescriptorTypeName](#) 14
[Icon](#) 14
[Id](#) 12
[IdentifierTypeName](#) 15
[Index](#) 14
[IsCached](#) 12
[IsDisplayed](#) 14
[IsOpenedInNewWindow](#) 14
[IsStatic](#) 19
[MetadataObjectType](#) 12
[MetadataRights](#) 18
[MethodInstanceType](#) 16

[Name](#) 12
[Position](#) 14
[SystemUtilityTypeName](#) 18
[TypeDescriptorTypeName](#) 17
[TypeReflectorTypeName](#) 17
[URL](#) 14
[ConnectionManagerTypeName field](#) 17
Count result set ([section 2.2.5.4](#) 21, [section 3.1.5.39.1](#) 59, [section 3.1.5.40.1](#) 60, [section 3.1.5.41.1](#) 60, [section 3.1.5.42.1](#) 61, [section 3.1.5.43.1](#) 62, [section 3.1.5.44.1](#) 62, [section 3.1.5.47.1](#) 64, [section 3.1.5.48.1](#) 64, [section 3.1.5.49.1](#) 65, [section 3.1.5.50.1](#) 65, [section 3.1.5.51.1](#) 66, [section 3.1.5.53.1](#) 67, [section 3.1.5.56.1](#) 69, [section 3.1.5.57.1](#) 70, [section 3.1.5.58.1](#) 70, [section 3.1.5.61.1](#) 72, [section 3.1.5.63.1](#) 73, [section 3.1.5.66.1](#) 74, [section 3.1.5.67.1](#) 75, [section 3.1.5.68.1](#) 75, [section 3.1.5.70.1](#) 76, [section 3.1.5.76.1](#) 79, [section 3.1.5.77.1](#) 80, [section 3.1.5.80.1](#) 81)
[Creating an Entity example](#) 100

D

Data model - abstract
[client](#) 98
[MetadataObject caching](#) 98
[server](#) 28
Data types
[common](#) 12
Data types - simple
[overview](#) 12
DataClass result set ([section 2.2.5.5](#) 21, [section 3.1.5.52.1](#) 67, [section 3.1.5.53.2](#) 67)
[Default Values result set](#) 68
[Deleting an Entity example](#) 102
[Direction field](#) 17

E

Entity result set ([section 2.2.5.6](#) 22, [section 3.1.5.56.2](#) 69, [section 3.1.5.57.2](#) 70, [section 3.1.5.58.2](#) 71, [section 3.1.5.59.1](#) 71)
[EntityId result set](#) 69
[EstimatedInstanceCount field](#) 12
Events
[local - client](#) 99
[local - server](#) 98
[timer - client](#) 99
[timer - server](#) 98
Examples
[cache invalidation](#) 103
[creating an Entity](#) 100
[deleting an Entity](#) 102
[overview](#) 100
[reading an Entity](#) 101
[reading the security information of a MetadataObject](#) 101

[updating an Entity](#) 102

F

Fields – common

[ConnectionManagerTypeName](#) 17

[Direction](#) 17

[EstimatedInstanceCount](#) 12

[FilterDescriptorTypeName](#) 14

[Icon](#) 14

[Id](#) 12

[IdentifierTypeName](#) 15

[Index](#) 14

[IsCached](#) 12

[IsDisplayed](#) 14

[IsOpenedInNewWindow](#) 14

[IsStatic](#) 19

[MetadataObjectType](#) 12

[MetadataRights](#) 18

[MethodInstanceType](#) 16

[Name](#) 12

[Position](#) 14

[SystemUtilityTypeName](#) 18

[TypeDescriptorTypeName](#) 17

[TypeReflectorTypeName](#) 17

[URL](#) 14

[Fields - vendor-extensible](#) 11

[FilterDescriptor result set](#) ([section 2.2.5.7](#) 22, [section 3.1.5.60.1](#) 71, [section 3.1.5.61.2](#) 72)

[FilterDescriptorTypeName field](#) 14

[Flag structures - overview](#) 19

G

[Glossary](#) 8

H

Higher-layer triggered events

[client](#) 99

[server](#) 29

I

[Icon field](#) 14

[Id field](#) 12

[Identifier result set](#) ([section 2.2.5.8](#) 23, [section 3.1.5.62.1](#) 72, [section 3.1.5.63.2](#) 73)

[IdentifierTypeName field](#) 15

[Implementer - security considerations](#) 105

[Index field](#) 14

[Index of security parameters](#) 105

[Informative references](#) 9

Initialization

[client](#) 99

[server](#) 29

Interfaces - client

[Metadata Client details](#) 98

[Introduction](#) 8

[IsCached field](#) 12

[IsDisplayed field](#) 14

[IsOpenedInNewWindow field](#) 14

[IsStatic field](#) 19

L

Local events

[client](#) 99

[server](#) 98

[Localized Name result set](#) 60

M

Message processing

[client](#) 99

[server](#) 29

Messages

[Access Control Entry result set](#) 58

[Action result set](#) 19

[ActionParameter result set](#) ([section 2.2.5.2](#) 20, [section 3.1.5.38.1](#) 59, [section 3.1.5.39.2](#) 59)

[Association result set](#) ([section 2.2.5.3](#) 20, [section 3.1.5.45.1](#) 63, [section 3.1.5.46.1](#) 63, [section 3.1.5.47.2](#) 64, [section 3.1.5.48.2](#) 64, [section 3.1.5.49.2](#) 65)

[binary structures](#) 19

[bit fields](#) 19

[Cache Version Stamps result set](#) 65

[common data types](#) 12

[Count result set](#) ([section 2.2.5.4](#) 21, [section 3.1.5.39.1](#) 59, [section 3.1.5.40.1](#) 60, [section 3.1.5.41.1](#) 60, [section 3.1.5.42.1](#) 61, [section 3.1.5.43.1](#) 62, [section 3.1.5.44.1](#) 62, [section 3.1.5.47.1](#) 64, [section 3.1.5.48.1](#) 64, [section 3.1.5.49.1](#) 65, [section 3.1.5.50.1](#) 65, [section 3.1.5.51.1](#) 66, [section 3.1.5.53.1](#) 67, [section 3.1.5.56.1](#) 69, [section 3.1.5.57.1](#) 70, [section 3.1.5.58.1](#) 70, [section 3.1.5.61.1](#) 72, [section 3.1.5.63.1](#) 73, [section 3.1.5.66.1](#) 74, [section 3.1.5.67.1](#) 75, [section 3.1.5.68.1](#) 75, [section 3.1.5.70.1](#) 76, [section 3.1.5.76.1](#) 79, [section 3.1.5.77.1](#) 80, [section 3.1.5.80.1](#) 81)

[DataClass result set](#) ([section 2.2.5.5](#) 21, [section 3.1.5.52.1](#) 67, [section 3.1.5.53.2](#) 67)

[Default Values result set](#) 68

[Entity result set](#) ([section 2.2.5.6](#) 22, [section 3.1.5.56.2](#) 69, [section 3.1.5.57.2](#) 70, [section 3.1.5.58.2](#) 71, [section 3.1.5.59.1](#) 71)

[EntityId result set](#) 69

[enumerations](#) 12

[FilterDescriptor result set](#) ([section 2.2.5.7](#) 22, [section 3.1.5.60.1](#) 71, [section 3.1.5.61.2](#) 72)

[flag structures](#) 19

[Identifier result set](#) ([section 2.2.5.8](#) 23, [section 3.1.5.62.1](#) 72, [section 3.1.5.63.2](#) 73)

[Localized Name result set](#) 60

[Method result set](#) ([section 2.2.5.9](#) 23, [section 3.1.5.64.1](#) 73, [section 3.1.5.68.2](#) 75)

[MethodInstance result set](#) ([section 2.2.5.10](#) 24, [section 3.1.5.65.1](#) 74, [section 3.1.5.66.2](#) 74, [section 3.1.5.67.2](#) 75)

[Parameter result set](#) ([section 2.2.5.11](#) 24, [section 3.1.5.69.1](#) 76, [section 3.1.5.70.2](#) 76)

[Property result set](#) 77

[result sets](#) 19
[simple data types](#) 12
[System Data result set](#) 78
 System Instance result set ([section 2.2.5.12](#) 25,
[section 3.1.5.42.2](#) 62, [section 3.1.5.43.2](#) 62,
[section 3.1.5.75.1](#) 79, [section 3.1.5.76.2](#) 79)
 System result set ([section 2.2.5.13](#) 25, [section](#)
[3.1.5.44.2](#) 62, [section 3.1.5.73.1](#) 78, [section](#)
[3.1.5.77.2](#) 80)
[table structures](#) 27
[transport](#) 12
 TypeDescriptor result set ([section 2.2.5.14](#) 26,
[section 3.1.5.51.2](#) 66, [section 3.1.5.72.1](#) 77,
[section 3.1.5.78.1](#) 80, [section 3.1.5.79.1](#) 81,
[section 3.1.5.80.2](#) 81)
[view structures](#) 27
[XML structures](#) 27
 Messages - common fields
[ConnectionManagerTypeName](#) 17
[Direction](#) 17
[EstimatedInstanceCount](#) 12
[FilterDescriptorTypeName](#) 14
[Icon](#) 14
[Id](#) 12
[IdentifierTypeName](#) 15
[Index](#) 14
[IsCached](#) 12
[IsDisplayed](#) 14
[IsOpenedInNewWindow](#) 14
[IsStatic](#) 19
[MetadataObjectType](#) 12
[MetadataRights](#) 18
[MethodInstanceType](#) 16
[Name](#) 12
[Position](#) 14
[SystemUtilityTypeName](#) 18
[TypeDescriptorTypeName](#) 17
[TypeReflectorTypeName](#) 17
[URL](#) 14
[Metadata Client details interface](#) 98
[MetadataObjectType field](#) 12
[MetadataRights field](#) 18
 Method result set ([section 2.2.5.9](#) 23, [section](#)
[3.1.5.64.1](#) 73, [section 3.1.5.68.2](#) 75)
 MethodInstance result set ([section 2.2.5.10](#) 24,
[section 3.1.5.65.1](#) 74, [section 3.1.5.66.2](#) 74,
[section 3.1.5.67.2](#) 75)
[MethodInstanceType field](#) 16
 Methods
[proc_ar_AddOrInsertLocalizedNameForMetadataOb](#)
[jectId](#) 29
[proc_ar_AddOrInsertPropertyForMetadataObjectI](#)
[d](#) 30
[proc_ar_BumpCacheInvalidationCounter](#) 31
[proc_ar_ClearAccessControlEntriesForMetadataOb](#)
[ject](#) 32
[proc_ar_CopyAccessControlEntriesForMetadataOb](#)
[jectId](#) 32
[proc_ar_CreateAction](#) 32
[proc_ar_CreateActionParameter](#) 34
[proc_ar_CreateAssociation](#) 34
[proc_ar_CreateEntity](#) 36
[proc_ar_CreateFilterDescriptor](#) 37
[proc_ar_CreateIdentifier](#) 37
[proc_ar_CreateMethod](#) 38
[proc_ar_CreateMethodInstance](#) 39
[proc_ar_CreateParameter](#) 41
[proc_ar_CreateSystem](#) 42
[proc_ar_CreateSystemInstance](#) 43
[proc_ar_CreateTypeDescriptor](#) 43
[proc_ar_DeleteActionById](#) 45
[proc_ar_DeleteActionParameterById](#) 46
[proc_ar_DeleteAssociationById](#) 46
[proc_ar_DeleteDefaultValue](#) 47
[proc_ar_DeleteEntityById](#) 48
[proc_ar_DeleteFilterDescriptorById](#) 49
[proc_ar_DeleteIdentifierById](#) 49
[proc_ar_DeleteLocalizedNameForMetadataObject](#)
[ByLCID](#) 50
[proc_ar_DeleteLocalizedNamesByMetadataObject](#)
[Id](#) 51
[proc_ar_DeleteMethodById](#) 51
[proc_ar_DeleteMethodInstanceById](#) 52
[proc_ar_DeleteParameterById](#) 53
[proc_ar_DeletePropertiesById](#) 54
[proc_ar_DeletePropertyForMetadataObjectId](#) 54
[proc_ar_DeleteSystemById](#) 55
[proc_ar_DeleteSystemInstanceById](#) 56
[proc_ar_DeleteTypeDescriptorById](#) 56
[proc_ar_EnsureApplicationRegistryExists](#) 57
[proc_ar_GetAccessControlEntriesForMetadataObj](#)
[ect](#) 57
[proc_ar_GetActionById](#) 58
[proc_ar_GetActionParameterById](#) 59
[proc_ar_GetActionParametersForActionWithCount](#)
[59](#)
[proc_ar_GetActionsForEntityWithCount](#) 60
[proc_ar_GetAllLocalizedNamesForMetadataObject](#)
[WithCount](#) 60
[proc_ar_GetAllSystemInstancesLikeNameWithCou](#)
[nt](#) 61
[proc_ar_GetAllSystemInstancesWithCount](#) 62
[proc_ar_GetAllSystemsWithCount](#) 62
[proc_ar_GetAssociationById](#) 62
[proc_ar_GetAssociationByName](#) 63
[proc_ar_GetAssociationsForDataClassWithCount](#)
[63](#)
[proc_ar_GetAssociationsForEntityAndRoleWithCo](#)
[unt](#) 64
[proc_ar_GetAssociationsForMethodWithCount](#) 65
[proc_ar_GetCacheInvalidationCountersWithCount](#)
[65](#)
[proc_ar_GetChildTypeDescriptorsForTypeDescript](#)
[orWithCount](#) 66
[proc_ar_GetDataClassById](#) 66
[proc_ar_GetDataClassesForSystemWithCount](#) 67
[proc_ar_GetDefaultValuesForTypeDescriptor](#) 67
[proc_ar_GetDependentEntitiesForEntity](#) 68
[proc_ar_GetEntitiesForAssociationAndRoleWithCo](#)
[unt](#) 69
[proc_ar_GetEntitiesForSystemLikeNameWithCoun](#)
[t](#) 69

[proc ar GetEntitiesForSystemWithCount](#) 70
[proc ar GetEntityById](#) 71
[proc ar GetFilterDescriptorById](#) 71
[proc ar GetFilterDescriptorsForMethodWithCount](#) 71
[proc ar GetIdentifierById](#) 72
[proc ar GetIdentifiersForEntityWithCount](#) 72
[proc ar GetMethodById](#) 73
[proc ar GetMethodInstanceById](#) 73
[proc ar GetMethodInstancesForDataClassWithCount](#) 74
[proc ar GetMethodInstancesForMethodWithCount](#) 74
[proc ar GetMethodsForDataClassWithCount](#) 75
[proc ar GetParameterById](#) 75
[proc ar GetParametersForMethodWithCount](#) 76
[proc ar GetPropertiesForMetadataObject](#) 76
[proc ar GetRootTypeDescriptorForParameter](#) 77
[proc ar GetSystemById](#) 77
[proc ar GetSystemDataBySystemName](#) 78
[proc ar GetSystemInstanceById](#) 78
[proc ar GetSystemInstancesForSystemWithCount](#) 79
[proc ar GetSystemsLikeNameWithCount](#) 79
[proc ar GetTypeDescriptorById](#) 80
[proc ar GetTypeDescriptorsByNameAndParameter](#) 80
[proc ar GetTypeDescriptorsForFilterDescriptorWithCount](#) 81
[proc ar SetAccessControlEntryForMetadataObject](#) 81
[proc ar SetDefaultAction](#) 82
[proc ar SetDefaultValuesForTypeDescriptor](#) 83
[proc ar SetSystemDataBySystemName](#) 84
[proc ar UpdateActionById](#) 84
[proc ar UpdateActionParameterById](#) 85
[proc ar UpdateAssociationById](#) 86
[proc ar UpdateEntityById](#) 87
[proc ar UpdateFilterDescriptorById](#) 88
[proc ar UpdateIdentifierById](#) 89
[proc ar UpdateMethodById](#) 90
[proc ar UpdateMethodInstanceById](#) 91
[proc ar UpdateParameterById](#) 93
[proc ar UpdateSystemById](#) 94
[proc ar UpdateSystemInstanceById](#) 95
[proc ar UpdateTypeDescriptorById](#) 96

N

[Name field](#) 12
[Normative references](#) 9

O

[Overview \(synopsis\)](#) 10

P

Parameter result set ([section 2.2.5.11](#) 24, [section 3.1.5.69.1](#) 76, [section 3.1.5.70.2](#) 76)
[Parameters - security index](#) 105
[Position field](#) 14

[Preconditions](#) 10
[Prerequisites](#) 10
[proc ar AddOrInsertLocalizedNameForMetadataObjectById method](#) 29
[proc ar AddOrInsertPropertyForMetadataObjectById method](#) 30
[proc ar BumpCacheInvalidationCounter method](#) 31
[proc ar ClearAccessControlEntriesForMetadataObject method](#) 32
[proc ar CopyAccessControlEntriesForMetadataObjectById method](#) 32
[proc ar createAction method](#) 32
[proc ar createActionParameter method](#) 34
[proc ar CreateAssociation method](#) 34
[proc ar CreateEntity method](#) 36
[proc ar CreateFilterDescriptor method](#) 37
[proc ar CreateIdentifier method](#) 37
[proc ar CreateMethod method](#) 38
[proc ar CreateMethodInstance method](#) 39
[proc ar CreateParameter method](#) 41
[proc ar CreateSystem method](#) 42
[proc ar CreateSystemInstance method](#) 43
[proc ar CreateTypeDescriptor method](#) 43
[proc ar DeleteActionById method](#) 45
[proc ar DeleteActionParameterById method](#) 46
[proc ar DeleteAssociationById method](#) 46
[proc ar DeleteDefaultValue method](#) 47
[proc ar DeleteEntityById method](#) 48
[proc ar DeleteFilterDescriptorById method](#) 49
[proc ar DeleteIdentifierById method](#) 49
[proc ar DeleteLocalizedNameForMetadataObjectByLCID method](#) 50
[proc ar DeleteLocalizedNamesByMetadataObjectById method](#) 51
[proc ar DeleteMethodById method](#) 51
[proc ar DeleteMethodInstanceById method](#) 52
[proc ar DeleteParameterById method](#) 53
[proc ar DeletePropertiesById method](#) 54
[proc ar DeletePropertyForMetadataObjectById method](#) 54
[proc ar DeleteSystemById method](#) 55
[proc ar DeleteSystemInstanceById method](#) 56
[proc ar DeleteTypeDescriptorById method](#) 56
[proc ar EnsureApplicationRegistryExists method](#) 57
[proc ar GetAccessControlEntriesForMetadataObject method](#) 57
[proc ar GetActionById method](#) 58
[proc ar GetActionParameterById method](#) 59
[proc ar GetActionParametersForActionWithCount method](#) 59
[proc ar GetActionsForEntityWithCount method](#) 60
[proc ar GetAllLocalizedNamesForMetadataObjectWithCount method](#) 60
[proc ar GetAllSystemInstancesLikeNameWithCount method](#) 61
[proc ar GetAllSystemInstancesWithCount method](#) 62
[proc ar GetAllSystemsWithCount method](#) 62
[proc ar GetAssociationById method](#) 62
[proc ar GetAssociationByName method](#) 63

[proc ar GetAssociationsForDataClassWithCount method](#) 63
[proc ar GetAssociationsForEntityAndRoleWithCount method](#) 64
[proc ar GetAssociationsForMethodWithCount method](#) 65
[proc ar GetCacheInvalidationCountersWithCount method](#) 65
[proc ar GetChildTypeDescriptorsForTypeDescriptor WithCount method](#) 66
[proc ar GetDataClassById method](#) 66
[proc ar GetDataClassesForSystemWithCount method](#) 67
[proc ar GetDefaultValuesForTypeDescriptor method](#) 67
[proc ar GetDependentEntitiesForEntity method](#) 68
[proc ar GetEntitiesForAssociationAndRoleWithCount method](#) 69
[proc ar GetEntitiesForSystemLikeNameWithCount method](#) 69
[proc ar GetEntitiesForSystemWithCount method](#) 70
[proc ar GetEntityById method](#) 71
[proc ar GetFilterDescriptorById method](#) 71
[proc ar GetFilterDescriptorsForMethodWithCount method](#) 71
[proc ar GetIdentifierById method](#) 72
[proc ar GetIdentifiersForEntityWithCount method](#) 72
[proc ar GetMethodById method](#) 73
[proc ar GetMethodInstanceById method](#) 73
[proc ar GetMethodInstancesForDataClassWithCount method](#) 74
[proc ar GetMethodInstancesForMethodWithCount method](#) 74
[proc ar GetMethodsForDataClassWithCount method](#) 75
[proc ar GetParameterById method](#) 75
[proc ar GetParametersForMethodWithCount method](#) 76
[proc ar GetPropertiesForMetadataObject method](#) 76
[proc ar GetRootTypeDescriptorForParameter method](#) 77
[proc ar GetSystemById method](#) 77
[proc ar GetSystemDataBySystemName method](#) 78
[proc ar GetSystemInstanceById method](#) 78
[proc ar GetSystemInstancesForSystemWithCount method](#) 79
[proc ar GetSystemsLikeNameWithCount method](#) 79
[proc ar GetTypeDescriptorById method](#) 80
[proc ar GetTypeDescriptorsByNameAndParameter method](#) 80
[proc ar GetTypeDescriptorsForFilterDescriptorWithCount method](#) 81
[proc ar SetAccessControlEntryForMetadataObject method](#) 81
[proc ar SetDefaultAction method](#) 82
[proc ar SetDefaultValuesForTypeDescriptor method](#) 83
[proc ar SetSystemDataBySystemName method](#) 84

[proc ar UpdateActionById method](#) 84
[proc ar UpdateActionParameterById method](#) 85
[proc ar UpdateAssociationById method](#) 86
[proc ar UpdateEntityById method](#) 87
[proc ar UpdateFilterDescriptorById method](#) 88
[proc ar UpdateIdentifierById method](#) 89
[proc ar UpdateMethodById method](#) 90
[proc ar UpdateMethodInstanceById method](#) 91
[proc ar UpdateParameterById method](#) 93
[proc ar UpdateSystemById method](#) 94
[proc ar UpdateSystemInstanceById method](#) 95
[proc ar UpdateTypeDescriptorById method](#) 96
[Product behavior](#) 106
[Property result set](#) 77

R

[Reading an Entity example](#) 101
[Reading the security information of a MetadataObject example](#) 101

References

[informative](#) 9
[normative](#) 9

[Relationship to other protocols](#) 10

Result sets - messages

[Access Control Entry](#) 58

[Action](#) 19

[ActionParameter](#) ([section 2.2.5.2](#) 20, [section 3.1.5.38.1](#) 59, [section 3.1.5.39.2](#) 59)

[Association](#) ([section 2.2.5.3](#) 20, [section 3.1.5.45.1](#) 63, [section 3.1.5.46.1](#) 63, [section 3.1.5.47.2](#) 64, [section 3.1.5.48.2](#) 64, [section 3.1.5.49.2](#) 65)

[Cache Version Stamps](#) 65

[Count](#) ([section 2.2.5.4](#) 21, [section 3.1.5.39.1](#) 59, [section 3.1.5.40.1](#) 60, [section 3.1.5.41.1](#) 60, [section 3.1.5.42.1](#) 61, [section 3.1.5.43.1](#) 62, [section 3.1.5.44.1](#) 62, [section 3.1.5.47.1](#) 64, [section 3.1.5.48.1](#) 64, [section 3.1.5.49.1](#) 65, [section 3.1.5.50.1](#) 65, [section 3.1.5.51.1](#) 66, [section 3.1.5.53.1](#) 67, [section 3.1.5.56.1](#) 69, [section 3.1.5.57.1](#) 70, [section 3.1.5.58.1](#) 70, [section 3.1.5.61.1](#) 72, [section 3.1.5.63.1](#) 73, [section 3.1.5.66.1](#) 74, [section 3.1.5.67.1](#) 75, [section 3.1.5.68.1](#) 75, [section 3.1.5.70.1](#) 76, [section 3.1.5.76.1](#) 79, [section 3.1.5.77.1](#) 80, [section 3.1.5.80.1](#) 81)

[DataClass](#) ([section 2.2.5.5](#) 21, [section 3.1.5.52.1](#) 67, [section 3.1.5.53.2](#) 67)

[Default Values](#) 68

[Entity](#) ([section 2.2.5.6](#) 22, [section 3.1.5.56.2](#) 69, [section 3.1.5.57.2](#) 70, [section 3.1.5.58.2](#) 71, [section 3.1.5.59.1](#) 71)

[EntityId](#) 69

[FilterDescriptor](#) ([section 2.2.5.7](#) 22, [section 3.1.5.60.1](#) 71, [section 3.1.5.61.2](#) 72)

[Identifier](#) ([section 2.2.5.8](#) 23, [section 3.1.5.62.1](#) 72, [section 3.1.5.63.2](#) 73)

[Localized Name](#) 60

[Method](#) ([section 2.2.5.9](#) 23, [section 3.1.5.64.1](#) 73, [section 3.1.5.68.2](#) 75)

[MethodInstance](#) ([section 2.2.5.10](#) 24, [section 3.1.5.65.1](#) 74, [section 3.1.5.66.2](#) 74, [section 3.1.5.67.2](#) 75)
[Parameter](#) ([section 2.2.5.11](#) 24, [section 3.1.5.69.1](#) 76, [section 3.1.5.70.2](#) 76)
[Property](#) 77
[System](#) ([section 2.2.5.13](#) 25, [section 3.1.5.44.2](#) 62, [section 3.1.5.73.1](#) 78, [section 3.1.5.77.2](#) 80)
[System Data](#) 78
[System Instance](#) ([section 2.2.5.12](#) 25, [section 3.1.5.42.2](#) 62, [section 3.1.5.43.2](#) 62, [section 3.1.5.75.1](#) 79, [section 3.1.5.76.2](#) 79)
[TypeDescriptor](#) ([section 2.2.5.14](#) 26, [section 3.1.5.51.2](#) 66, [section 3.1.5.72.1](#) 77, [section 3.1.5.78.1](#) 80, [section 3.1.5.79.1](#) 81, [section 3.1.5.80.2](#) 81)
[Result sets - overview](#) 19
 Result sets – server
 Action ([section 3.1.5.37.1](#) 59, [section 3.1.5.40.2](#) 60)

S

Security
 [implementer considerations](#) 105
 [parameter index](#) 105
 Sequencing rules
 [client](#) 99
 [server](#) 29
 Server
 [abstract data model](#) 28
 Action result set ([section 3.1.5.37.1](#) 59, [section 3.1.5.40.2](#) 60)
 [higher-layer triggered events](#) 29
 [initialization](#) 29
 [local events](#) 98
 [message processing](#) 29
 [proc ar AddOrInsertLocalizedNameForMetadataObjectById method](#) 29
 [proc ar AddOrInsertPropertyForMetadataObjectById method](#) 30
 [proc ar BumpCacheInvalidationCounter method](#) 31
 [proc ar ClearAccessControlEntriesForMetadataObject method](#) 32
 [proc ar CopyAccessControlEntriesForMetadataObjectById method](#) 32
 [proc ar createAction method](#) 32
 [proc ar createActionParameter method](#) 34
 [proc ar CreateAssociation method](#) 34
 [proc ar CreateEntity method](#) 36
 [proc ar CreateFilterDescriptor method](#) 37
 [proc ar CreateIdentifier method](#) 37
 [proc ar CreateMethod method](#) 38
 [proc ar CreateMethodInstance method](#) 39
 [proc ar CreateParameter method](#) 41
 [proc ar CreateSystem method](#) 42
 [proc ar CreateSystemInstance method](#) 43
 [proc ar CreateTypeDescriptor method](#) 43
 [proc ar DeleteActionById method](#) 45
 [proc ar DeleteActionParameterById method](#) 46

[proc ar DeleteAssociationById method](#) 46
 [proc ar DeleteDefaultValue method](#) 47
 [proc ar DeleteEntityById method](#) 48
 [proc ar DeleteFilterDescriptorById method](#) 49
 [proc ar DeleteIdentifierById method](#) 49
 [proc ar DeleteLocalizedNameForMetadataObjectByLCID method](#) 50
 [proc ar DeleteLocalizedNamesByMetadataObjectById method](#) 51
 [proc ar DeleteMethodById method](#) 51
 [proc ar DeleteMethodInstanceById method](#) 52
 [proc ar DeleteParameterById method](#) 53
 [proc ar DeletePropertiesById method](#) 54
 [proc ar DeletePropertyForMetadataObjectById method](#) 54
 [proc ar DeleteSystemById method](#) 55
 [proc ar DeleteSystemInstanceById method](#) 56
 [proc ar DeleteTypeDescriptorById method](#) 56
 [proc ar EnsureApplicationRegistryExists method](#) 57
 [proc ar GetAccessControlEntriesForMetadataObject method](#) 57
 [proc ar GetActionById method](#) 58
 [proc ar GetActionParameterById method](#) 59
 [proc ar GetActionParametersForActionWithCount method](#) 59
 [proc ar GetActionsForEntityWithCount method](#) 60
 [proc ar GetAllLocalizedNamesForMetadataObjectWithCount method](#) 60
 [proc ar GetAllSystemInstancesLikeNameWithCount method](#) 61
 [proc ar GetAllSystemInstancesWithCount method](#) 62
 [proc ar GetAllSystemsWithCount method](#) 62
 [proc ar GetAssociationById method](#) 62
 [proc ar GetAssociationByName method](#) 63
 [proc ar GetAssociationsForDataClassWithCount method](#) 63
 [proc ar GetAssociationsForEntityAndRoleWithCount method](#) 64
 [proc ar GetAssociationsForMethodWithCount method](#) 65
 [proc ar GetCacheInvalidationCountersWithCount method](#) 65
 [proc ar GetChildTypeDescriptorsForTypeDescriptorOrWithCount method](#) 66
 [proc ar GetDataClassById method](#) 66
 [proc ar GetDataClassesForSystemWithCount method](#) 67
 [proc ar GetDefaultValuesForTypeDescriptor method](#) 67
 [proc ar GetDependentEntitiesForEntity method](#) 68
 [proc ar GetEntitiesForAssociationAndRoleWithCount method](#) 69
 [proc ar GetEntitiesForSystemLikeNameWithCount method](#) 69
 [proc ar GetEntitiesForSystemWithCount method](#) 70
 [proc ar GetEntityById method](#) 71

- [proc ar GetFilterDescriptorById method](#) 71
- [proc ar GetFilterDescriptorsForMethodWithCount method](#) 71
- [proc ar GetIdentifierById method](#) 72
- [proc ar GetIdentifiersForEntityWithCount method](#) 72
- [proc ar GetMethodById method](#) 73
- [proc ar GetMethodInstanceById method](#) 73
- [proc ar GetMethodInstancesForDataClassWithCount method](#) 74
- [proc ar GetMethodInstancesForMethodWithCount method](#) 74
- [proc ar GetMethodsForDataClassWithCount method](#) 75
- [proc ar GetParameterById method](#) 75
- [proc ar GetParametersForMethodWithCount method](#) 76
- [proc ar GetPropertiesForMetadataObject method](#) 76
- [proc ar GetRootTypeDescriptorForParameter method](#) 77
- [proc ar GetSystemById method](#) 77
- [proc ar GetSystemDataBySystemName method](#) 78
- [proc ar GetSystemInstanceById method](#) 78
- [proc ar GetSystemInstancesForSystemWithCount method](#) 79
- [proc ar GetSystemsLikeNameWithCount method](#) 79
- [proc ar GetTypeDescriptorById method](#) 80
- [proc ar GetTypeDescriptorsByNameAndParameter method](#) 80
- [proc ar GetTypeDescriptorsForFilterDescriptorWithCount method](#) 81
- [proc ar SetAccessControlEntryForMetadataObject method](#) 81
- [proc ar SetDefaultAction method](#) 82
- [proc ar SetDefaultValuesForTypeDescriptor method](#) 83
- [proc ar SetSystemDataBySystemName method](#) 84
- [proc ar UpdateActionById method](#) 84
- [proc ar UpdateActionParameterById method](#) 85
- [proc ar UpdateAssociationById method](#) 86
- [proc ar UpdateEntityById method](#) 87
- [proc ar UpdateFilterDescriptorById method](#) 88
- [proc ar UpdateIdentifierById method](#) 89
- [proc ar UpdateMethodById method](#) 90
- [proc ar UpdateMethodInstanceById method](#) 91
- [proc ar UpdateParameterById method](#) 93
- [proc ar UpdateSystemById method](#) 94
- [proc ar UpdateSystemInstanceById method](#) 95
- [proc ar UpdateTypeDescriptorById method](#) 96
- [sequencing rules](#) 29
- [timer events](#) 98
- [timers](#) 29
- Simple data types
 - [overview](#) 12
- [Standards assignments](#) 11
- Structures
 - [binary](#) 19

- [table and view](#) 27
- [XML](#) 27
- [System Data result set](#) 78
- System Instance result set ([section 2.2.5.12](#) 25, [section 3.1.5.42.2](#) 62, [section 3.1.5.43.2](#) 62, [section 3.1.5.75.1](#) 79, [section 3.1.5.76.2](#) 79)
- System result set ([section 2.2.5.13](#) 25, [section 3.1.5.44.2](#) 62, [section 3.1.5.73.1](#) 78, [section 3.1.5.77.2](#) 80)
- [SystemUtilityTypeName field](#) 18

T

- [Table structures - overview](#) 27
- Timer events
 - [client](#) 99
 - [server](#) 98
- Timers
 - [client](#) 99
 - [server](#) 29
- [Tracking changes](#) 111
- [Transport](#) 12
- Triggered events - higher-layer
 - [client](#) 99
 - [server](#) 29
- TypeDescriptor result set ([section 2.2.5.14](#) 26, [section 3.1.5.51.2](#) 66, [section 3.1.5.72.1](#) 77, [section 3.1.5.78.1](#) 80, [section 3.1.5.79.1](#) 81, [section 3.1.5.80.2](#) 81)
- [TypeDescriptorTypeName field](#) 17
- [TypeReflectorTypeName field](#) 17

U

- [Updating an Entity example](#) 102
- [URL field](#) 14

V

- [Vendor-extensible fields](#) 11
- [Versioning](#) 11
- [View structures - overview](#) 27

X

- [XML structures](#) 27