

# [MS-EMMSTORE]: Enterprise Metadata Service Database Schema

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto: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.

**Preliminary Documentation.** This Open Specification provides documentation for past and current releases and/or for the pre-release (beta) version of this technology. This Open Specification is final

documentation for past or current releases as specifically noted in the document, as applicable; it is preliminary documentation for the pre-release (beta) versions. Microsoft will release final documentation in connection with the commercial release of the updated or new version of this technology. As the documentation may change between this preliminary version and the final version of this technology, there are risks in relying on preliminary documentation. To the extent that you incur additional development obligations or any other costs as a result of relying on this preliminary documentation, you do so at your own risk.

## Revision Summary

Date	Revision History	Revision Class	Comments
07/13/2009	0.1	Major	Initial Availability
08/28/2009	0.2	Editorial	Revised and edited the technical content
11/06/2009	0.3	Editorial	Revised and edited the technical content
02/19/2010	1.0	Major	Updated and revised the technical content
03/31/2010	1.01	Editorial	Revised and edited the technical content
04/30/2010	1.02	Minor	Updated the technical content
06/07/2010	1.03	Editorial	Revised and edited the technical content
06/29/2010	1.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	1.05	Minor	Clarified the meaning of the technical content.
09/27/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	1.05	No change	No changes to the meaning, language, or formatting of the technical content.
01/20/2012	2.0	Major	Significantly changed the technical content.
04/11/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/16/2012	2.0	No change	No changes to the meaning, language, or formatting of the technical content.

# Table of Contents

<b>1 Introduction</b>	<b>8</b>
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	10
1.7 Versioning and Capability Negotiation	11
1.8 Vendor-Extensible Fields	11
1.9 Standards Assignments	11
<b>2 Messages</b>	<b>12</b>
2.1 Transport	12
2.2 Common Data Types	12
2.2.1 Simple Data Types and Enumerations	12
2.2.2 Bit Fields and Flag Structures	12
2.2.3 Binary Structures	12
2.2.4 Result Sets	12
2.2.4.1 DoesPartitionDataExistResultSet	12
2.2.4.2 GetChangeTimeResultSet	12
2.2.4.3 proc_ECM_GetGroupByGuid.ResultSet2	12
2.2.4.4 proc_ECM_GetGroupByGuid.ResultSet1	13
2.2.4.5 proc_ECM_GetGroupByGuid.ResultSet0	13
2.2.4.6 GetPackageResultSet	14
2.2.4.7 proc_ECM_GetPackagesInformation.ResultSet0	14
2.2.4.8 proc_ECM_GetServiceSettings.ResultSet0	14
2.2.4.9 proc_ECM_GetSessionData.ResultSet5	14
2.2.4.10 proc_ECM_GetSessionData.ResultSet2	15
2.2.4.11 proc_ECM_GetSessionData.ResultSet1	15
2.2.4.12 proc_ECM_GetSessionData.ResultSet0	15
2.2.4.13 proc_ECM_MoveTermSetMembership.ResultSet0	16
2.2.4.14 proc_ECM_GetTerms.proc_ECM_GetTerms.IncludeTermSet.ResultSet5	16
2.2.4.15 proc_ECM_GetTerms.ResultSet4	16
2.2.4.16 proc_ECM_GetTerms.ResultSet3	17
2.2.4.17 proc_ECM_GetTerms.ResultSet2	17
2.2.4.18 proc_ECM_GetTerms.ResultSet1	18
2.2.4.19 proc_ECM_GetTerms.ResultSet0	18
2.2.4.20 proc_ECM_GetTermSetByGuid.ResultSet4	19
2.2.4.21 proc_ECM_GetTermSetByGuid.ResultSet3	19
2.2.4.22 proc_ECM_RetrieveApplicationLog.ResultSet0	20
2.2.4.23 proc_ECM_RetrieveTableNames.ResultSet0	21
2.2.4.24 proc_ECM_GetChangesForListSync.ResultSet0	21
2.2.4.25 proc_ECM_GetChanges.proc_ECM_GetChanges.Default.ResultSet0	22
2.2.4.26 proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMTermLabel.ResultSet0	23

2.2.4.27	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMUsedTerms.ResultSet0 .....	23
2.2.4.28	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMPackage.ResultSet0 .....	24
2.2.4.29	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMTermSet.ResultSet0 .....	24
2.2.4.30	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMTerm.ResultSet0 .....	25
2.2.4.31	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMMergedTerm.ResultSet0 .....	26
2.2.4.32	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMTermProperty.ResultSet0 .....	26
2.2.4.33	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMServiceSettings.ResultSet0 .....	27
2.2.4.34	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMTermDescription.ResultSet0 .....	27
2.2.4.35	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMTermSetMembership.ResultSet0 .....	28
2.2.4.36	proc_ECM_SelectPartitionData.proc_ECM_SelectPartitionData.ECMPermission.ResultSet0 .....	28
2.2.4.37	proc_ECM_GetSessionData.ResultSet4 .....	29
2.2.4.38	proc_ECM_GetSessionData.ResultSet6 .....	29
2.2.4.39	proc_ECM_GetSortedChildTermIds.ResultSet0 .....	29
2.2.4.40	proc_ECM_GetChangedTermSets.ResultSet0 .....	29
2.2.4.41	proc_ECM_SearchAllTermSetsByProperty.ResultSet0 .....	30
2.2.4.42	proc_ECM_GetTermSetByGuid.ResultSet5 .....	30
2.2.4.43	proc_ECM_GetTerms.LcidNotNull.ResultSet9 .....	30
2.2.4.44	proc_ECM_SearchByLabel.Optimized.ResultSet5 .....	32
2.2.5	Tables and Views .....	32
2.2.5.1	ECMApplicationLog .....	32
2.2.5.2	ECMChangeLog .....	33
2.2.5.3	ECMTermSet .....	34
2.2.5.4	ECMPackage .....	35
2.2.5.5	ECMTerm .....	36
2.2.5.6	ECMTermLabel .....	37
2.2.5.7	ECMLanguage .....	37
2.2.5.8	ECMUsedTerms .....	37
2.2.5.9	ECMMergedTerm .....	38
2.2.5.10	ECMTermProperty .....	38
2.2.5.11	ECMGroup .....	39
2.2.5.12	ECMServiceSettings .....	39
2.2.5.13	ECMTermDescription .....	40
2.2.5.14	ECMTermSetMembership .....	40

2.2.5.15	ECMPermission .....	41
2.2.6	XML Structures .....	42
2.2.6.1	Namespaces .....	42
2.2.6.2	Simple Types .....	42
2.2.6.3	Complex Types.....	42
2.2.6.4	Elements .....	42
2.2.6.5	Attributes .....	42
2.2.6.6	Groups .....	42
2.2.6.7	Attribute Groups.....	42
2.2.7	Raised Errors .....	42
<b>3</b>	<b>Protocol Details .....</b>	<b>46</b>
3.1	Server Details .....	46
3.1.1	Abstract Data Model .....	46
3.1.2	Timers .....	47
3.1.3	Initialization .....	47
3.1.4	Higher-Layer Triggered Events.....	47
3.1.5	Message Processing Events and Sequencing Rules.....	47
3.1.5.1	proc_ECM_AddTermLabel .....	48
3.1.5.2	proc_ECM_AddApplicationLogEntry .....	49
3.1.5.3	proc_ECM_AddLanguage .....	50
3.1.5.4	proc_ECM_AddTermDescription .....	50
3.1.5.5	proc_ECM_AddTermProperty .....	51
3.1.5.6	proc_ECM_AddTermSetMembership .....	52
3.1.5.7	proc_ECM_AddUsedTerm .....	53
3.1.5.8	proc_ECM_ClearPartitionData .....	54
3.1.5.9	proc_ECM_CreateGroup.....	55
3.1.5.10	proc_ECM_CreatePermission .....	55
3.1.5.11	proc_ECM_CreateTerm .....	56
3.1.5.12	proc_ECM_CreateTermSet .....	57
3.1.5.13	proc_ECM_DeleteGroup .....	59
3.1.5.14	proc_ECM_DeleteLanguage .....	60
3.1.5.15	proc_ECM_DeletePermission.....	60
3.1.5.16	proc_ECM_DeleteServiceSettings.....	61
3.1.5.17	proc_ECM_DeleteTerm .....	62
3.1.5.18	proc_ECM_DeleteTermDescription.....	62
3.1.5.19	proc_ECM_DeleteTermLabel .....	63
3.1.5.20	proc_ECM_DeleteTermProperty .....	63
3.1.5.21	proc_ECM_DeleteTermSet.....	63
3.1.5.22	proc_ECM_DoesPartitionDataExist.....	64
3.1.5.23	proc_ECM_GetAllTerms.....	65
3.1.5.24	proc_ECM_GetChanges.....	65
3.1.5.25	proc_ECM_GetChangeTime .....	66
3.1.5.26	proc_ECM_GetDescendentTerms.....	67
3.1.5.27	proc_ECM_GetGroupByGuid .....	68
3.1.5.28	proc_ECM_GetGroups.....	68
3.1.5.29	proc_ECM_GetPackage .....	69
3.1.5.30	proc_ECM_GetPackagesInformation .....	69
3.1.5.31	proc_ECM_GetServiceSettings.....	70
3.1.5.32	proc_ECM_GetSessionData .....	70
3.1.5.33	proc_ECM_GetSortedChildTermIds.....	71
3.1.5.34	proc_ECM_GetTermInTermSetByGuid.....	71
3.1.5.35	proc_ECM_GetTerms .....	72

3.1.5.36	proc_ECM_GetTermSets .....	75
3.1.5.37	proc_ECM_GetTermSetsByLabel .....	77
3.1.5.38	proc_ECM_ImportPartitionData .....	78
3.1.5.39	proc_ECM_LogChange .....	79
3.1.5.40	proc_ECM_MergeTerm .....	79
3.1.5.41	proc_ECM_MoveTermSetMembership .....	81
3.1.5.42	proc_ECM_PopulateLabelsForDefaultLanguage .....	82
3.1.5.43	proc_ECM_RemoveApplicationLog .....	83
3.1.5.44	proc_ECM_RemoveUsedTermSets .....	83
3.1.5.45	proc_ECM_RetrieveApplicationLog .....	83
3.1.5.46	proc_ECM_RetrieveTableNames .....	84
3.1.5.47	proc_ECM_SearchByLabel .....	84
3.1.5.48	proc_ECM_SearchByProperty .....	87
3.1.5.49	proc_ECM_SearchTermSetByName .....	88
3.1.5.50	proc_ECM_SelectPartitionData .....	88
3.1.5.51	proc_ECM_SetDefaultLanguage .....	91
3.1.5.52	proc_ECM_SetPackage .....	92
3.1.5.53	proc_ECM_SetServiceSettings .....	92
3.1.5.54	proc_ECM_UpdateGroup .....	93
3.1.5.55	proc_ECM_UpdatePermission .....	94
3.1.5.56	proc_ECM_UpdateTerm .....	95
3.1.5.57	proc_ECM_UpdateTermDescription .....	96
3.1.5.58	proc_ECM_UpdateTermProperty .....	96
3.1.5.59	proc_ECM_UpdateTermSet .....	97
3.1.5.60	proc_ECM_UpdateTermSetMembership .....	99
3.1.5.61	proc_ECM_ClearUsedTerms .....	100
3.1.5.62	proc_ECM_GetChangesForListSync .....	101
3.1.5.63	proc_ECM_ImportPackage .....	101
3.1.5.64	proc_ECM_AddChangeLog .....	102
3.1.5.65	proc_ECM_GetChangesForFullListSync .....	103
3.1.5.66	proc_ECM_LogRestore .....	103
3.1.5.67	proc_ECM_UnpublishAllPackages .....	103
3.1.5.68	proc_ECM_GetChangedTermSets .....	104
3.1.5.69	proc_ECM_SearchTermSetsByProperty .....	104
3.1.5.70	proc_ECM_SearchAllTermSetsByProperty .....	105
3.1.6	Timer Events .....	106
3.1.7	Other Local Events .....	106
3.2	Client Details .....	106
3.2.1	Abstract Data Model .....	106
3.2.2	Timers .....	106
3.2.3	Initialization .....	106
3.2.4	Higher-Layer Triggered Events .....	106
3.2.5	Message Processing Events and Sequencing Rules .....	106
3.2.6	Timer Events .....	106
3.2.7	Other Local Events .....	106
<b>4</b>	<b>Protocol Examples .....</b>	<b>107</b>
4.1	Search Terms by a Label .....	107
4.2	Create a Group .....	108
<b>5</b>	<b>Security .....</b>	<b>109</b>
5.1	Security Considerations for Implementers .....	109
5.2	Index of Security Parameters .....	109

<b>6</b>	<b>Appendix A: Product Behavior .....</b>	<b>110</b>
<b>7</b>	<b>Change Tracking.....</b>	<b>111</b>
<b>8</b>	<b>Index .....</b>	<b>112</b>

Preliminary

# 1 Introduction

This document specifies the Enterprise Metadata Service Database Schema. This protocol enables protocol clients to perform operations against a store containing metadata. Typical operations include creating, deleting, merging, reusing, deprecating, describing, and translating metadata items.

Sections 1.8, 2, and 3 of this specification are normative and can contain the terms MAY, SHOULD, MUST, MUST NOT, and SHOULD NOT as defined in RFC 2119. Sections 1.5 and 1.9 are also normative but cannot contain those terms. All other sections and examples in this specification are informative.

## 1.1 Glossary

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

**access control list (ACL)**  
**Augmented Backus-Naur Form (ABNF)**  
**Coordinated Universal Time (UTC)**  
**GUID**  
**language code identifier (LCID)**  
**Universal Naming Convention (UNC)**

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

**content database**  
**content type**  
**content type package**  
**deprecated term**  
**descendant**  
**group**  
**internal identifier**  
**keyword**  
**locale**  
**merged term**  
**orphaned term**  
**partition**  
**partition identifier**  
**result set**  
**return code**  
**reused term**  
**security principal**  
**service application**  
**site**  
**site collection**  
**source term**  
**term**  
**term label**  
**term set**  
**term set group**  
**term store**  
**user name**  
**Web application**  
**XML namespace**

The following terms are specific to this document:

**content type syndication:** A process in which a published content type is made available for use in one or more additional site collections.

**open term set:** A term set to which new terms can be submitted by users of a tagging application.

**published content type:** A content type that was copied from one site collection and made available for use in another site collection.

**target term:** The term that persists after two terms are merged.

**term label path:** A string that contains the labels for a term and the labels for all of its parent terms.

**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

References to Microsoft Open Specifications documentation do not include a publishing year because links are to the latest version of the technical documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 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](mailto: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.

[Iseminger] 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://www.microsoft.com/mspress/books/5001.aspx>

[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-EMMWCF] Microsoft Corporation, "[Metadata Shared Service WCF Service Protocol Specification](#)".

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

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

[XMLNS] Bray, T., Hollander, D., Layman, A., et al., Eds., "Namespaces in XML 1.0 (Third Edition)", W3C Recommendation, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208/>

### 1.2.2 Informative References

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

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

### 1.3 Protocol Overview (Synopsis)

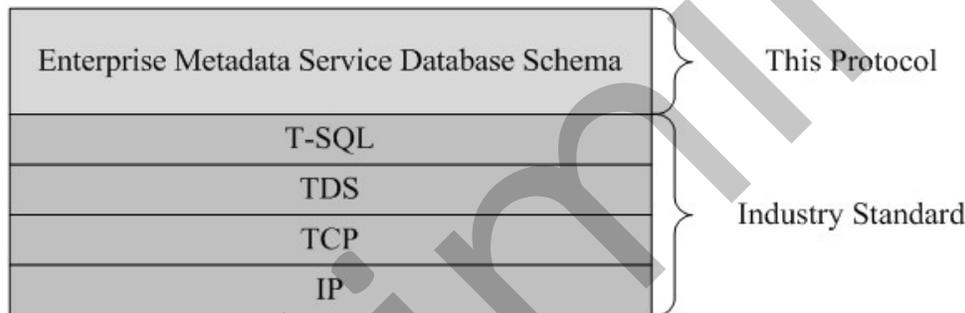
Enterprises wanting to interact with digital assets and resources by way of metadata applied to the item often find that the inability to manage, share, update or restrict metadata values from one or many centralized services in the enterprise makes it so that items have inconsistent or unreliable metadata.

This protocol enables protocol clients to create, delete, merge, reuse, deprecate, describe, translate and define synonyms for metadata **terms** to be applied to digital assets and resources. Typical scenarios include building hierarchical structures of terms (**term sets**). Each term in the hierarchy can have many **term labels** to represent synonyms of the term. A term can also be translated and for each supported language have a collection of synonyms. Once hierarchies are created they can be consumed by clients in their entirety or fragments of the term sets. The term sets are typically used as constrained lists from which users or application can select values to apply to a digital asset or resource.

To facilitate this, the **term store** contains **groups** of term sets. Each group allows for different user **access control lists (ACLs)**, and each term set is a hierarchy of terms. Terms can coexist in different location of the hierarchy within different term sets. Term sets also allow for custom sort order as well as unique settings for availability of a term or entire term set. This allows for greater flexibility when different divisions or sections of an enterprise have specific and differing expectations and needs of the metadata hierarchies.

### 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 which 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 term store database server.

### 1.6 Applicability Statement

This protocol is used to query, create and manage a term store, as well as groups, term sets, and terms.

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

### **1.7 Versioning and Capability Negotiation**

This document covers versioning issues in the following areas:

- **Structure versions:** This document specifies the only version of this protocol.
- **Localization:** This document specifies no locale-specific processes or data.

### **1.8 Vendor-Extensible Fields**

None.

### **1.9 Standards Assignments**

None.

## 2 Messages

### 2.1 Transport

[\[MS-TDS\]](#) specifies the transport protocol used to call the stored procedures, query SQL tables, get **return codes**, and return **result sets**.

### 2.2 Common Data Types

#### 2.2.1 Simple Data Types and Enumerations

#### 2.2.2 Bit Fields and Flag Structures

#### 2.2.3 Binary Structures

#### 2.2.4 Result Sets

##### 2.2.4.1 DoesPartitionDataExistResultSet

This stored procedure is called to create an access control list (ACL) for the term store or **term set group**. It creates an entry in the table **ECMPermission** (section [2.2.5.15](#)). The **ACL** assigns the user one or more of the following rights: none, edit term, edit term set, edit term group, add term set, edit permissions, manage term store, contributor, term group manager, term store administrator. The **Initialization** routine specified in step 2 of section [3.1.5](#) **MUST** be executed before calling this stored procedure.

```
UnnamedColumn0 int,
```

**UnnamedColumn0:** 0 if term store **partition** does not exist. Any other value indicates that the term store partition exists.

##### 2.2.4.2 GetChangeTimeResultSet

The **proc\_ECM\_GetChangeTime** result set **MUST** return a single row. The fields of the result set are populated using values from the **ECMChangeLog** table (section [2.2.5.2](#)). The T-SQL syntax for the result is as follows:

```
ChangeTime datetime,
```

**ChangeTime:** The UTC time when the change happened. The latest change date and time value, in **UTC**. ChangeTime **MUST NOT** be NULL.

##### 2.2.4.3 proc\_ECM\_GetGroupByGuid.ResultSet2

The **proc\_ECM\_GetGroupByGuid.ResultSet2** result set contains information about which term sets belong to a group. The fields are retrieved from the **ECMTermSet** table (section [2.2.5.3](#)). The T-SQL syntax for the result set is:

```
GroupId int,  
Id int,
```

**GroupId:** The **internal identifier (1)** of the group to which the term set belongs.

**Id:** The internal identifier (1) of the term set.

#### 2.2.4.4 **proc\_ECM\_GetGroupByGuid.ResultSet1**

The **proc\_ECM\_GetGroupByGuid.ResultSet1** result set contains information about which permissions on a group have been assigned to various **security principals**. The fields are retrieved from the **ECMPermission** table (section [2.2.5.15](#)). The T-SQL syntax for the result set is:

```
GroupId int,  
PrincipalName nvarchar(255),  
Rights bigint,
```

**GroupId:** The internal identifier (1) of a group.

**PrincipalName:** The user NT account name, in Domain\Account format.

**Rights:** Rights. The value is a bitmask that specifies rights of the user. Possible values are defined in the Rights field of the ECMPermission table.

#### 2.2.4.5 **proc\_ECM\_GetGroupByGuid.ResultSet0**

The **proc\_ECM\_GetGroupByGuid.ResultSet0** result set contains information about a single group. The fields are retrieved from the **ECMGroup** table (section [2.2.5.11](#)). The T-SQL syntax for the result set is:

```
Id nvarchar(255),  
PartitionId uniqueidentifier,  
UniqueId uniqueidentifier,  
Name nvarchar(255),  
Description nvarchar(1000),  
LastModifiedTime datetime,  
CreatedTime datetime,  
Type int,
```

**Id:** The identifier of the group.

**PartitionId:** The unique identifier of the term store partition to which the group belongs.

**UniqueId:** The unique identifier of the group.

**Name:** The name of the group.

**Description:** The description of the group.

**LastModifiedTime:** The UTC time of last modification to the group.

**CreatedTime:** The UTC time when the group was created.

**Type:** The type of group. Possible values are defined in the ECMGroup table definition in the Type field.

#### 2.2.4.6 GetPackageResultSet

The **proc\_ECM\_GetPackage.ResultSet0** result set contains information about a single **content type** package. The fields are retrieved from **ECMPackage** table (section [2.2.5.4](#)). The T-SQL syntax for the result set is:

```
ContentSize int,  
Content varbinary(max),
```

**ContentSize:** The size of the content type package content, in bytes.

**Content:** The content of the content type package. The format is described in the **Content** field of the **ECMPackage** table.

#### 2.2.4.7 proc\_ECM\_GetPackagesInformation.ResultSet0

The **proc\_ECM\_GetPackagesInformation.ResultSet0** result set contains information about content type packages that have been modified since the specified last update time. The fields are retrieved from **ECMPackage** table (section [2.2.5.4](#)). The T-SQL syntax for the result set is as follows:

```
Id varchar(512),  
Type uniqueidentifier,  
LastModifiedTime datetime,  
IsPublished bit,
```

**Id:** The identifier of the content type.

**Type:** The identifier of the type of object in the package. The value MUST be "B4AD3A44-D934-4C91-8D1F-463ACEADE443".

**LastModifiedTime:** The UTC time of the last modification to the content type.

**IsPublished:** Specifies whether the content type is published.

#### 2.2.4.8 proc\_ECM\_GetServiceSettings.ResultSet0

This result set contains the service setting data as XML specified by the [\[MS-EMMWCF\]](#) **ServiceSettingsSettingsXmlDoc** complex type. The T-SQL syntax for the result set is as follows:

```
Settings nvarchar(max),
```

**Settings:** XML that contains the term store identifier, the **content type syndication** hub URL, and whether content type syndication error reporting is enabled.

#### 2.2.4.9 proc\_ECM\_GetSessionData.ResultSet5

The **proc\_ECM\_GetSessionData.ResultSet5** result set MUST return one or more rows of GUID identifiers which identify term set, term, or group items in the term store. The fields of the result set are populated using values from the **ECMTermSet** table (section [2.2.5.3](#)), or the **ECMTerm** table (section [2.2.5.5](#)), or the **ECMGroup** table (section [2.2.5.11](#)). The T-SQL syntax for the result set is as follows:

UniqueId uniqueidentifier,

**UniqueId:** The unique identifier for a term set, term, or group in the term store. It MUST NOT be NULL.

#### 2.2.4.10 `proc_ECM_GetSessionData.ResultSet2`

The `proc_ECM_GetSessionData.ResultSet2` result set MUST return a row for every term store specific permission in the specified term store partition or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMPermission** table (section [2.2.5.15](#)). The T-SQL syntax for the result set is as follows:

```
PrincipalName nvarchar(255),  
Rights bigint,
```

**PrincipalName:** The user NT account name, in Domain\Account format. It MUST NOT be NULL.

**Rights:** Rights. The value is a bitmask that specifies rights of the user. Possible values are defined in the **Rights** field of the **ECMPermission** table. It MUST NOT be NULL.

#### 2.2.4.11 `proc_ECM_GetSessionData.ResultSet1`

The `proc_ECM_GetSessionData.ResultSet1` result set MUST return a row for every group in the specified term store partition or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMGroup** table (section [2.2.5.11](#)). The T-SQL syntax for the result set is as follows:

```
Id int,
```

**Id:** The identifier of the group. It MUST NOT be NULL.

#### 2.2.4.12 `proc_ECM_GetSessionData.ResultSet0`

The `proc_ECM_GetSessionData.ResultSet0` result set MUST return a row for every language in the specified term store partition or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMLanguage** table (section [2.2.5.7](#)). The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
WorkingLanguageId int,  
IsDefaultLanguage bit,
```

**PartitionId:** The identifier of the term store partition that the language belongs to. It MUST NOT be NULL.

**WorkingLanguageId:** The identifier of the corresponding locale for the language. It MUST NOT be NULL.

**IsDefaultLanguage:** Specifies whether the language is the default language for the term store partition. It MUST NOT be NULL.

#### 2.2.4.13 **proc\_ECM\_MoveTermSetMembership.ResultSet0**

The **proc\_ECM\_MoveTermSetMembership.ResultSet0** result set MUST return a row for every homograph matching the search condition or zero rows if there is no match. The fields of the result set are populated using values from the **ECMTermLabel** table (section [2.2.5.6](#)). The T-SQL syntax for the result is as follows:

```
Label nvarchar(255),  
LCID int,
```

**Label:** The label of the homograph term.

**LCID:** The **language code identifier (LCID)** of the language that the label is in.

#### 2.2.4.14 **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.IncludeTermSet.ResultSet5**

The **proc\_ECM\_GetTerms.ResultSet5** result set MUST return a row for every immediate child term of a term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMTermSetMembership** table (section [2.2.5.14](#)). The T-SQL syntax for the result set is as follows:

```
ParentTermId int,  
TermSetId int,  
ChildTermId int,
```

**ParentTermId:** The internal identifier (1) of the parent term of the current term in the term set. If the term is at the root of the term set, this will be 0.

**TermSetId:** The internal identifier (1) of the term set of the term membership.

**ChildTermId:** The internal identifier (1) of the child term in the term set.

#### 2.2.4.15 **proc\_ECM\_GetTerms.ResultSet4**

The **proc\_ECM\_GetTerms.ResultSet4** result set MUST return a row for every custom property of a term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMTermProperty** table (section [2.2.5.10](#)). The T-SQL syntax for the result set is as follows:

```
TermId int,  
TermSetId int,  
PropertyName nvarchar(255),  
PropertyValue nvarchar(max),
```

**TermId:** The internal identifier (1) of the term to which the property belongs.

**TermSetId:** The internal identifier (1) of the term set of the term membership [<1>](#).

**PropertyName:** The name of the term property.

**PropertyValue:** The value of the term property.

### 2.2.4.16 proc\_ECM\_GetTerms.ResultSet3

The **proc\_ECM\_GetTerms.ResultSet3** result set MUST return a row for every parent of a term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMTermSetMembership** table (section [2.2.5.14](#)). The T-SQL syntax for the result set is as follows:

```
TermSetId int,  
TermId int,  
IdPath nvarchar(884),  
ParentTermId int,  
AvailableForTagging bit,  
CustomSortOrder nvarchar(max),  
IsSource bit,  
PinSourceTermSetId int,
```

**TermSetId:** The internal identifier (1) of the term set of the term membership.

**TermId:** The internal identifier (1) of the child term in the term set.

**IdPath:** Specifies the hierarchy (path) of the term in the term set. The value is composed of a sequence of term internal identifiers (1) separated by a '\\' character. The first internal identifier is for the root term, the second is for the child of the root term, and so on until the last internal identifier (1) which is for the term itself.

**ParentTermId:** The internal identifier (1) of the parent term of the current term in the term set. If the term is at the root of the term set, this is zero.

**AvailableForTagging:** Specifies if the term is available for tagging when used in the context of the given term set.

**CustomSortOrder:** **ECMTermSetMembership:CustomSortOrder.**

**IsSource:** For terms that are reused in multiple term sets, one instance is designated as the **source term**, whose permissions are used for operations that affect all instances of the term. If the term has only one instance, then that instance is considered the source term.

**PinSourceTermSetId:** The internal identifier (1) of the source term set that this term is pinned to. If the term is not pinned to any term set, this MUST be NULL.

### 2.2.4.17 proc\_ECM\_GetTerms.ResultSet2

The **proc\_ECM\_GetTerms.ResultSet2** result set MUST return a row for every description of a term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMTermDescription** table (section [2.2.5.13](#)). The T-SQL syntax for the result set is as follows:

```
TermId int,  
LCID int,  
Description nvarchar(1000),
```

**TermId:** The internal identifier (1) of the term the description is for.

**LCID:** The identifier of the locale that the description is for.

**Description:** The description of the term.

#### 2.2.4.18 `proc_ECM_GetTerms.ResultSet1`

The `proc_ECM_GetTerms.ResultSet1` result set MUST return a row for every term label of a term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the `ECMTermLabel` table (section [2.2.5.6](#)). The T-SQL syntax for the result set is as follows:

```
TermId int,  
LCID int,  
Label nvarchar(255),  
IsDefault bit,
```

**TermId:** The internal identifier (1) of the term to which the label belongs.

**LCID:** The identifier of the locale that the label is for.

**Label:** The text of the label.

**IsDefault:** Specifies whether this label is the default label for the given LCID. A given term can only have one default label per LCID.

#### 2.2.4.19 `proc_ECM_GetTerms.ResultSet0`

The `proc_ECM_GetTerms.ResultSet0` result set MUST return a row for every term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the `ECMTerm` table (section [2.2.5.5](#)). The T-SQL syntax for the result set is as follows:

```
Id int,  
PartitionId uniqueidentifier,  
CreatedTime datetime,  
LastModifiedTime datetime,  
Owner nvarchar(255),  
UniqueId uniqueidentifier,  
LastUsedTime datetime,  
UseCount int,  
IsDeprecated bit,  
IsDeleted bit,  
MergedIdList nvarchar(max),
```

**Id:** The internal identifier (1) of the term.

**PartitionId:** The identifier of the term store partition to which the term belongs.

**CreatedTime:** The UTC time when the term was created.

**LastModifiedTime:** The UTC time of the last modification to the term object.

**Owner:** The name of the owner of the term.

**UniqueId:** The unique identifier of the term.

**LastUsedTime:** Reserved, MUST be ignored.

**UseCount:** Reserved, MUST be 0.

**IsDeprecated:** Specifies whether the term has been deprecated.

**IsDeleted:** Specifies whether the term has been deleted.

**MergedIdList:** A list of term identifiers separated by a backslash (\) that represent all of the terms that have been merged into the current term.

#### 2.2.4.20 **proc\_ECM\_GetTermSetByGuid.ResultSet4**

The **proc\_ECM\_GetTermSetByGuid.ResultSet4** result set contains the root terms of one or more term sets. The result set MUST return one row for every root term in each term set. The T-SQL syntax for the result set is as follows:

```
TermSetId int,  
TermId int,
```

**TermSetId:** The internal identifier (1) of the term set.

**TermId:** The internal identifier (1) of the term.

#### 2.2.4.21 **proc\_ECM\_GetTermSetByGuid.ResultSet3**

The **proc\_ECM\_GetTermSetByGuid.ResultSet3** result set contains the properties of a term set. The result set MUST return one row for each term set. The T-SQL syntax for the result set is as follows:

```
Id int,  
PartitionId uniqueidentifier,  
CreatedTime datetime,  
LastModifiedTime datetime,  
Owner nvarchar(255),  
CustomSortOrder nvarchar(max),  
UniqueId uniqueidentifier,  
Name nvarchar(max),  
Description nvarchar(1000),  
Type tinyint,  
IsOpen bit,  
AvailableForTagging bit,  
Stakeholders nvarchar(1000),  
Contact nvarchar(320),  
GroupId int,
```

**Id:** The internal identifier (1) of the term set.

**PartitionId:** The identifier of the term store partition to which the term set belongs.

**CreatedTime:** The UTC time when the term set was created.

**LastModifiedTime:** The UTC time of the last modification to the term set object.

**Owner:** The name of the owner of the term set.

**CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST

be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid **GUID** from the **UniqueId** column of the **ECMTerm** table (section [2.2.5.5](#)).

**UniqueId:** The unique identifier of the term set.

**Name:** The name of the term set.

**Description:** The description of the term set.

**Type:** The type of the term set. This value **MUST** be one of the values described in the following table.

Value	Description
0	A normal term set.
1	The keyword term set.
2	The term set that stores <b>orphaned terms</b> .

**IsOpen:** Specifies whether a term set allows users without explicit rights to the group to add terms to the term set.

**AvailableForTagging:** Specifies whether a term set should be visible to users that are not group administrators or term store administrators.

**Stakeholders:** A list of users that have an interest in the term set.

**Contact:** The user or group that should be contacted if a user wants to request the addition of a term to the term set, and the **IsOpen** flag of the term set is **false**.

**GroupId:** The internal identifier (1) of the group to which the term set belongs.

#### 2.2.4.22 **proc\_ECM\_RetrieveApplicationLog.ResultSet0**

The **proc\_ECM\_RetrieveApplicationLog.ResultSet0** result set **MUST** return a row for every entry in the content type syndication application log for the specified term store partition or 0 rows if there is no match. The fields of the result set are populated using values from **ECMApplicationLog** table (section [2.2.5.1](#)). The T-SQL syntax for the result set is as follows:

```
SiteUrl nvarchar(2000),  
ObjectName nvarchar(max),  
Stage nvarchar(100),  
Message nvarchar(max),  
CreatedTime datetime,
```

**SiteUrl:** The URL of a site in which the log entry is generated.

**ObjectName:** The name of the object for which the log entry is generated.

**Stage:** The name of the content type syndication stage at which the log entry is generated. This is defined in the **Stage** field of the **ECMApplicationLog** table.

**Message:** The content of the log entry.

**CreatedTime:** The UTC time when the log entry is generated.

### 2.2.4.23 **proc\_ECM\_RetrieveTableNames.ResultSet0**

The **proc\_ECM\_RetrieveTableNames.ResultSet0** result set contains the names of all the user-defined tables in the database. The names of all the tables listed in [2.2.5](#) MUST be returned in this result set. The T-SQL syntax for the result set is as follows:

```
name nvarchar(128),
```

**name:** The name of a user-defined table in the database.

### 2.2.4.24 **proc\_ECM\_GetChangesForListSync.ResultSet0**

The **proc\_ECM\_GetChangesForListSync.ResultSet0** result contains MUST return a row for every change matching the search condition or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMUsedTerms** table (section [2.2.5.8](#)), the **ECMChangeLog** table (section [2.2.5.2](#)), and the **ECMServiceSettings** table (section [2.2.5.12](#)). The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
SiteGuid uniqueidentifier,  
TermSetId uniqueidentifier,  
GroupUniqueId uniqueidentifier,  
TermSetUniqueId uniqueidentifier,  
ObjectUniqueId uniqueidentifier,  
ObjectId int,  
ObjectType int,  
ChangeType int,  
ChangeTime datetime,  
ChangeData nvarchar(max),  
Settings nvarchar(max),
```

**PartitionId:** The term store partition where term is used. It MUST NOT be NULL.

**SiteGuid:** The identifier of the **site collection** where term is used. It MUST NOT be NULL.

**TermSetId:** The identifier of the term set to which used term belongs.

**GroupUniqueId:** The unique identifier of a group. The value MUST be NULL if the change is not scoped inside a group, such as language change, permission change, partition data import, partition data deletion, and used term change.

**TermSetUniqueId:** The unique identifier of a term set. The value MUST be NULL if the change is not scoped inside a term set, such as language change, permission change, partition data import, partition data deletion, group change, or used term change.

**ObjectUniqueId:** The unique identifier of the object that the change is for. The value MUST be NULL if the change is not for a specific object, such as language change, permission change, partition data import, partition data deletion, or used term change.

**ObjectId:** The internal identifier (1) of the object for which the change is.

**ObjectType:** The type of the object as defined by the **ObjectType** field of the **ECMChangeLog** table. It MUST NOT be NULL.

**ChangeType:** The type of the change as defined by the **ChangeType** field of the **ECMChangeLog** table. It MUST NOT be NULL.

**ChangeTime:** The UTC time when the change happened. It MUST NOT be NULL.

**ChangeData:** The details of the change. It MUST NOT be NULL.

**Settings:** XML that specifies the term store identifier, the content type syndication hub URL, and whether content type syndication error reporting is enabled or not. The XML MUST conform to the [\[MS-EMMWCF\]](#) **ServiceSettingsSettingsXmlDoc** complex type. It MUST NOT be NULL.

#### 2.2.4.25 **proc\_ECM\_GetChanges.proc\_ECM\_GetChanges.Default.ResultSet0**

The **proc\_ECM\_GetChanges.proc\_ECM\_GetChanges.Default.ResultSet0** result set MUST return a row for every change matching the search condition or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMChangeLog** table (section [2.2.5.2](#)). The T-SQL syntax for the result is as follows:

```
PartitionId uniqueidentifier,  
GroupUniqueId uniqueidentifier,  
TermSetUniqueId uniqueidentifier,  
ObjectUniqueId uniqueidentifier,  
ObjectId int,  
ObjectType int,  
ChangeType int,  
ChangeTime datetime,  
ChangeData nvarchar(max),  
ModifiedBy nvarchar(255),
```

**PartitionId:** The unique identifier of the term store partition. The value MUST NOT be NULL.

**GroupUniqueId:** The unique identifier of a group. The value MUST be NULL if the change is not scoped inside a group, such as language change, permission change, partition data import, partition data deletion, and used term change.

**TermSetUniqueId:** The unique identifier of a term set. The value MUST be NULL if the change is not scoped inside a term set, such as language change, permission change, partition data import, partition data deletion, group change, or used term change.

**ObjectUniqueId:** The unique identifier of the object that the change is for. The value MUST be NULL if the change is not for a specific object, such as language change, permission change, partition data import, partition data deletion, or used term change.

**ObjectId:** The internal identifier (1) of the object that the change is for.

**ObjectType:** The type of the object as defined by the **ObjectType** field of the **ECMChangeLog** table. It MUST NOT be NULL.

**ChangeType:** The type of the change as defined by the **ChangeType** field of the **ECMChangeLog** table. It MUST NOT be NULL.

**ChangeTime:** The UTC time when the change happened. It MUST NOT be NULL.

**ChangeData:** The details of the change.

**ModifiedBy:** Reserved. The contents are not specified and MUST be ignored by the protocol client.

#### 2.2.4.26

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermLabel.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermLabel.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMTermLabel** table (section [2.2.5.6](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
TermId int,  
LCID int,  
Label nvarchar(255),  
IsDefault bit,
```

**PartitionId:** The identifier of the term store partition to which the label's term belongs.

**TermId:** The internal identifier (1) of the term to which the label belongs.

**LCID:** The identifier of the locale that the label is for.

**Label:** The text of the label.

**IsDefault:** Specifies whether this label is the default label for the given LCID. A given term can only have one default label per LCID.

#### 2.2.4.27

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMUsedTerms.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMUsedTerms.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMUsedTerms** table (section [2.2.5.8](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
WebAppId uniqueidentifier,  
ContentDatabaseId uniqueidentifier,  
SiteGuid uniqueidentifier,  
TermSetId uniqueidentifier,  
TermId int,
```

**PartitionId:** The identifier of the term store partition to which the term belongs.

**WebAppId:** The identifier of the Web application which contains the site collection in which the term is being used.

**ContentDatabaseId:** The identifier of the content database for the site collection in which the term is being used.

**SiteGuid:** The identifier of the site collection in which the term is being used.

**TermSetId:** The unique identifier of the term set to which the term belongs.

**TermId:** The internal identifier (1) of the term.

#### 2.2.4.28

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMPackage.ResultSet0**

The **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMPackage.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMPackage** table (section [2.2.5.4](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
Id nvarchar(512),  
Type uniqueidentifier,  
IsPublished bit,  
LastModifiedTime datetime,  
ContentSize int,  
Content image,
```

**PartitionId:** The identifier of the term store partition to which the content type belongs.

**Id:** The identifier of the content type.

**Type:** The identifier of the type of object in the package. The value MUST be "B4AD3A44-D934-4C91-8D1F-463ACEADE443".

**IsPublished:** Specifies whether the content type is published.

**LastModifiedTime:** The UTC time of the last modification to the content type.

**ContentSize:** The size of the content type package content, in bytes.

**Content:** The content of the content type package. The format is described in the **Content** field of the **ECMPackage** table.

#### 2.2.4.29

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermSet.ResultSet0**

The **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermSet.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMTermSet** table (section [2.2.5.3](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
Id int,  
PartitionId uniqueidentifier,  
CreatedTime datetime,  
LastModifiedTime datetime,  
UniqueId uniqueidentifier,  
Owner nvarchar(255),  
CustomSortOrder nvarchar(max),  
Name nvarchar(max),  
Description nvarchar(1000),  
Type tinyint,  
IsOpen bit,  
AvailableForTagging bit,  
Stakeholders nvarchar(1000),  
Contact nvarchar(320),  
GroupId int,
```

**Id:** The internal identifier (1) of the term set.

**PartitionId:** The identifier of the term store partition to which the term set belongs.

**CreatedTime:** The UTC time when the term set was created.

**LastModifiedTime:** The UTC time of the last modification to the term set object.

**UniqueId:** The unique identifier of the term set.

**Owner:** The name of the owner of the term set.

**CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the **UniqueId** column of the **ECMTerm** table (section [2.2.5.5](#)).

**Name:** The name of the term set.

**Description:** The description of the term set.

**Type:** The type of the term set as defined in the **Type** field of the **ECMTermSet** table.

**IsOpen:** Specifies whether a term set allows users without explicit rights to the group to add terms to the term set.

**AvailableForTagging:** Specifies whether a term set should be visible to users that are not group administrators or term store administrators.

**Stakeholders:** A list of users that have an interest in the term set.

**Contact:** The user or group that should be contacted if a user wants to request the addition of a term to the term set, and the **IsOpen** flag of the term set is **false**.

**GroupId:** The internal identifier (1) of the group to which the term set belongs.

#### 2.2.4.30

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTerm.ResultSet0**

The **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTerm.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMTerm** table (section [2.2.5.5](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
Id int,
PartitionId uniqueidentifier,
CreatedTime datetime,
LastModifiedTime datetime,
UniqueId uniqueidentifier,
LastUsedTime datetime,
UseCount int,
Owner nvarchar(255),
IsDeprecated bit,
IsDeleted bit,
MergedIdList varchar(max),
```

**Id:** The internal identifier (1) of the term.

**PartitionId:** The identifier of the term store partition to which the term belongs.

**CreatedTime:** The UTC time when the term was created.

**LastModifiedTime:** The UTC time of the last modification to the term object.

**UniqueId:** The unique identifier of the term.

**LastUsedTime:** Reserved.

**UseCount:** Reserved.

**Owner:** The name of the owner of the term.

**IsDeprecated:** Specifies whether the term has been deprecated.

**IsDeleted:** Specifies whether the term has been deleted.

**MergedIdList:** A list of term identifiers separated by a backslash (\) that represent all of the terms that have been merged into the current term.

#### 2.2.4.31

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMMergedTerm.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMMergedTerm.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMMergedTerm** table (section [2.2.5.9](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
TermUniqueId uniqueidentifier,  
TargetTermId int,
```

**PartitionId:** The identifier of the term store partition to which the merged term belongs.

**TermUniqueId:** The unique identifier of the merged term.

**TargetTermId:** The internal identifier (1) of the target term into which the term has been merged.

#### 2.2.4.32

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermProperty.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermProperty.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMTermProperty** table (section [2.2.5.10](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
TermId int,  
PropertyName nvarchar(255),  
PropertyValue nvarchar(255),
```

**PartitionId:** The unique identifier of the term store partition to which the term belongs.

**TermId:** The internal identifier (1) of the term to which the property belongs.

**PropertyName:** The name of the term property.

**PropertyValue:** The value of the term property.

#### 2.2.4.33

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMServiceSettings.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMServiceSettings.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMServiceSettings** table (section [2.2.5.12](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
Settings ntext,
```

**PartitionId:** The unique identifier of the partition to which the settings belong.

**Settings:** Xml that specifies the term store identifier, the content type syndication hub URL, and whether content type syndication error reporting is enabled. The XML MUST conform to the **complexType** [\[MS-EMMWCF\] ServiceSettingsSettingsXmlDoc](#).

#### 2.2.4.34

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermDescription.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermDescription.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMTermDescription** table (section [2.2.5.13](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
TermId int,  
LCID int,  
Description nvarchar(1000),
```

**PartitionId:** The unique identifier of the term store partition to which the term belongs.

**TermId:** The internal identifier (1) of the term that the description is for.

**LCID:** The identifier of the locale that the description is for.

**Description:** The description of the term.

#### 2.2.4.35

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermSetMembership.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMTermSetMembership.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMTermSetMembership** table (section [2.2.5.14](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
TermSetId int,  
TermId int,  
Path nvarchar(884),  
ParentTermId int,  
AvailableForTagging bit,  
CustomSortOrder ntext,  
IsSource bit,
```

**PartitionId:** The identifier of the term store partition to which the term hierarchy belongs.

**TermSetId:** The internal identifier (1) of the term set that the term membership is in.

**TermId:** The internal identifier (1) of the child term in the term set.

**Path:** Specifies the hierarchy (path) of the term in the term set. The value is composed of a sequence of term internal identifiers (1) separated by a backslash (\). The first internal identifier (1) is for the root term, the second is for the child of the root term, and so on until the last internal identifier (1), which is for the term itself.

**ParentTermId:** The internal identifier (1) of the parent term of the current term in the term set. If the term is at the root of the term set, this will be zero.

**AvailableForTagging:** Specifies if the term is available for tagging when used in the context of the given term set.

**CustomSortOrder: ECMTermSetMembership:CustomSortOrder.**

**IsSource:** For terms that are reused in multiple term sets, one instance is designated as the source term, whose permissions are used for operations that affect all instances of the term. If the term has only one instance, then that instance is considered the source term.

#### 2.2.4.36

##### **proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMPermission.ResultSet0**

The

**proc\_ECM\_SelectPartitionData.proc\_ECM\_SelectPartitionData.ECMPermission.ResultSet0** result set MUST return all rows and corresponding fields from the **ECMPermission** table (section [2.2.5.15](#)) for the specified term store partition. The T-SQL syntax for the result set is as follows:

```
PartitionId uniqueidentifier,  
GroupId int,  
PrincipalName nvarchar(255),  
DisplayName nvarchar(255),
```

```
RawSid varbinary(512),
Rights bigint,
```

**PartitionId:** The identifier of the term store partition to which the group belongs.

**GroupId:** The internal identifier (1) of a group.

**PrincipalName:** The user NT account name, in Domain\Account format.

**DisplayName:** The users display name.

**RawSid:** Reserved. The value must be NULL.

**Rights:** Rights. The value is a bitmask that specifies rights of the user. Possible values are defined in the **Rights** field of the **ECMPermission** table.

#### 2.2.4.37 **proc\_ECM\_GetSessionData.ResultSet4**

The **proc\_ECM\_GetSessionData.ResultSet4** result set MUST return a single row containing the identifier of the **keywords** term set. The fields of the result set are populated using values from the **ECMTermSet** table (section [2.2.5.3](#)). The T-SQL syntax for the result set is as follows:

```
UniqueId uniqueidentifier,
```

**UniqueId:** **ECMTermSet:UniqueId** for the term set that contains all of the terms that have been used as keywords that are not in existing **termsets**. It MUST NOT be NULL.

#### 2.2.4.38 **proc\_ECM\_GetSessionData.ResultSet6**

The **proc\_ECM\_GetSessionData.ResultSet6** result set MUST return a single row containing the UTC time of the most recent entry in the **ECMChangeLog** table (section [2.2.5.2](#)).

```
ChangeTime datetime,
```

**ChangeTime:** The UTC time of the most recent entry to the **ECMChangeLog** table.

#### 2.2.4.39 **proc\_ECM\_GetSortedChildTermIds.ResultSet0**

The **proc\_ECM\_GetSortedChildTermIds.ResultSet0** contains a set of term identifiers. The identifiers are retrieved from the **ECMTerm** table (section [2.2.5.5](#)). The T-SQL syntax for the result set is as follows:

```
UniqueId uniqueidentifier,
```

**UniqueId:** The unique identifier of the term.

#### 2.2.4.40 **proc\_ECM\_GetChangedTermSets.ResultSet0**

The **proc\_ECM\_GetChangedTermSets.ResultSet0** result set MUST return a row for every term set that has been changed since the requested time that matches the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMTermSet** table (section [2.2.5.14](#)). The T-SQL syntax for the result set is as follows<2>

```
PartitionId uniqueidentifier,  
TermSetUniqueId uniqueidentifier,  
MaxChangeTime datetime,
```

**PartitionId:** The internal identifier of the term set.

**TermSetUniqueId:** The unique identifier of the term set.

**MaxChangeTime:** The UTC time when the most recent change happened in the term store. The latest change date and time value, in UTC. This change time can be passed in to subsequent calls to **proc\_ECM\_GetChangedTermSets** as the starting time to check for future changes.

#### 2.2.4.41 **proc\_ECM\_SearchAllTermSetsByProperty.ResultSet0**

```
PartitionId uniqueidentifier,  
TermSetId int,  
PropertyName nvarchar(255),  
PropertyValue nvarchar(max),
```

**PartitionId:** Type the Column description.

**TermSetId:** Type the Column description.

**PropertyName:** Type the Column description.

**PropertyValue:** Type the Column description.

#### 2.2.4.42 **proc\_ECM\_GetTermSetByGuid.ResultSet5**

The **proc\_ECM\_GetTermSetByGuid.ResultSet5** result set MUST return a row for every custom property of the term set. The fields of the result set are populated using values from the **ECMTermProperty** table (section [2.2.5.10](#)). The T-SQL syntax for the result set is as follows<3>:

```
TermId int,  
TermSetId int,  
PropertyName nvarchar(255),  
PropertyValue nvarchar(max),
```

**TermId:** The value MUST be zero.

**TermSetId:** The internal identifier (1) of the term Set.

**PropertyName:** The name of the term set property.

**PropertyValue:** The value of the term set property.

#### 2.2.4.43 **proc\_ECM\_GetTerms.LcidNotNull.ResultSet9**

The **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.LcidNotNull.ResultSet9** result set MUST return one or more rows for each term set membership matching the search condition, or 0 rows if there is no match. A term set membership consists of a term and a term set that it is a member of. A term may be a member of multiple term sets, so each membership MUST return one row in this result set. The fields of the result set are populated using values from the **ECMTermSetMembership**

(section [2.2.5.14](#)), **ECMTerm** (section [2.2.5.5](#)), and **ECMTermSet** tables (section [2.2.5.3](#)). The T-SQL syntax for the result set is as follows:

```
TermSetId int,  
TermId int,  
IdPath nvarchar(884),  
ParentTermId int,  
AvailableForTagging bit,  
CustomSortOrder nvarchar(max),  
IsSource bit,  
PinSourceTermSetId int,  
Path nvarchar(max),  
FullGuidPath nvarchar(max),  
ParentTermGuid uniqueidentifier,  
TermSetName nvarchar(max),  
TermSetGuid uniqueidentifier,  
Type tinyint,
```

**TermSetId:** The internal identifier (1) of the term set to which the term belongs.

**TermId:** The internal identifier (1) of the term.

**IdPath:** Specifies the hierarchy (path) of the term in the term set. The value is composed of a sequence of term internal identifiers (1) separated by a backslash (\). The first internal identifier (1) is for the root term, the second is for the child of the root term, and so on until the last internal identifier (1), which is for the term itself.

**ParentTermId:** The internal identifier (1) of the parent term of the term identified by TermId. If the term is at the root of the term set, the **ParentTermId** is zero.

**AvailableForTagging:** Specifies if the term is available for tagging when used in the context of the term set identified by **TermSetId**. A term may be available for tagging in one term set but not another.

**CustomSortOrder:** **ECMTermSetMembership:CustomSortOrder.**

**IsSource:** For terms that appear in multiple term sets, only one term set will contain the source term. This field identifies the whether this this term set contains the source term. The term set containing the source term is used to check for permissions when editing the term.

**PinSourceTermSetId:** The internal identifier (1) of the source term set that this term is pinned to. If the term is not pinned to any term set, this MUST be NULL. [<4>](#)

**Path:** This value indicates the hierarchy of the term matching the search condition. It is composed of pairs of values separated by a vertical bar (|). Each pair describes a level of the hierarchy and is composed of the internal identifier (1) of a term, followed by a semicolon (;), followed by the text of the label. The first pair is for the ancestor term at the top of the hierarchy. The last pair is for the term matching the search criteria. If the term matching the search criteria is a root term then this field MUST contain only one pair. The term label that is retrieved for each member of the hierarchy is determined based on the locale identifier specified in the search.

**FullGuidPath:** A set of identifiers separated by the vertical bar (|), which indicates the hierarchy of the term matching the search condition. The first identifier is that of the group, followed by that of the term set, followed by the identifier of each ancestor term in the hierarchy of the search term, and finally the identifier of the search term corresponding to **TermId**.

**ParentTermGuid:** The identifier of the parent term of the term matching the search criteria, or NULL if the search term is a root term.

**TermSetName:** The name of the term set identified by **TermSetId**.

**TermSetGuid:** The identifier of the term set identified by **TermSetId**.

**Type:** The type of the term set identified by **TermSetId**.

#### 2.2.4.44 **proc\_ECM\_SearchByLabel.Optimized.ResultSet5**

The **proc\_ECM\_SearchByLabel.Optimized.ResultSet5** result set MUST return a row for every custom property of a term matching the search condition, or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMTermProperty** table (section [2.2.5.10](#)). The T-SQL syntax for the result set is as follows

```
TermId int,  
TermSetId int,  
PropertyName nvarchar(255),  
PropertyValue nvarchar(max),
```

**TermId:** The internal identifier (1) of the term to which the property belongs.

**TermSetId:** The internal identifier (1) of the term set of the term membership. [<5>](#)

**PropertyName:** The name of the term property.

**PropertyValue:** The value of the term property.

### 2.2.5 Tables and Views

#### 2.2.5.1 **ECMApplicationLog**

This table is used to store the application log entries that are generated during content type syndication and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
SiteUrl nvarchar(2000) NOT NULL,  
ObjectName nvarchar(max) NULL,  
Stage varchar(100) NOT NULL,  
Message nvarchar(max) NOT NULL,  
CreatedTime datetime NOT NULL,
```

**PartitionId:** The identifier of the term store partition to which content types have been published.

**SiteUrl:** The URL of a site in which the log entry is generated.

**ObjectName:** The name of the object for which the log entry is generated.

**Stage:** The name of the content type syndication stage at which the log entry is generated.

Possible parameter values:

Value	Description
<b>None</b>	Not started yet.
<b>Publishing</b>	Publishing a content type.
<b>TimeStampRetrieving</b>	Retrieving the time stamp from a content type subscriber site.
<b>PackageRetrieving</b>	Retrieving a content type package from term store.
<b>Importing</b>	Importing a package on a content type subscriber site.
<b>ImportingContentType</b>	Importing a content type package on a content type subscriber site.
<b>ImportingNonContentType</b>	Reserved.
<b>ContentTypeUnsharing</b>	Marking a content type as not syndicated on a content type subscriber site.
<b>PreCheck</b>	Checking prerequisites before importing on a content type subscriber site.
<b>SiteRefreshing</b>	Retrieving a content type subscriber site.
<b>WorkflowImporting</b>	Importing the workflows associated with a content type on a content type subscriber site..
<b>ContentTypePolicyDirtyBagUpdating</b>	Updating the policy dirty bag of a content type on a content type subscriber site.
<b>ContentTypeSealing</b>	Making a content type read-only.
<b>TimeStampUpdating</b>	Updating the time stamp on a content type subscriber site.
<b>LogTrimming</b>	Trimming the content type publishing error log list to prevent it from getting too big.
<b>LogSynchronizing</b>	Retrieving the content type syndication application log.
<b>Unpublishing</b>	Unpublishing a content type.
<b>Failed</b>	Content type syndication has failed to complete.
<b>Succeeded</b>	Content type syndication has succeeded.

**Message:** The content of the log entry.

**CreatedTime:** The UTC time when the log entry is generated.

### 2.2.5.2 ECMChangeLog

This table is used to store all of the changes that have happened in the term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,
GroupUniqueId uniqueidentifier NULL,
TermSetUniqueId uniqueidentifier NULL,
ObjectUniqueId uniqueidentifier NULL,
ObjectId int NULL,
```

```
ObjectType int NOT NULL,  
ChangeType int NOT NULL,  
ChangeTime datetime NOT NULL,  
ChangeData nvarchar(max) NULL,  
ModifiedBy nvarchar(255) NULL,
```

**PartitionId:** The identifier of the term store partition to which the change belongs.

**GroupUniqueId:** The unique identifier of a group. The value MUST be NULL if the change is not scoped inside a group, such as language change, permission change, partition data import, partition data deletion, and used term change.

**TermSetUniqueId:** The unique identifier of a term set. The value MUST be NULL if the change is not scoped inside a term set, such as language change, permission change, partition data import, partition data deletion, group change, or used term change.

**ObjectUniqueId:** The unique identifier of the object that the change is for. The value MUST be NULL if the change is not for a specific object, such as language change, permission change, partition data import, partition data deletion, or used term change.

**ObjectId:** The internal identifier (1) of the object that the change is for.

**ObjectType:** The type of the object.

**ChangeType:** The type of the change.

**ChangeTime:** The UTC time when the change happened.

**ChangeData:** The details of the change.

**ModifiedBy:** Reserved. The value MUST be NULL.

### 2.2.5.3 ECMTermSet

This table is used to store all of the term sets in the term store and is defined as follows.

```
Id int NOT NULL,  
PartitionId uniqueidentifier NOT NULL,  
CreatedTime datetime NOT NULL,  
LastModifiedTime datetime NULL,  
UniqueId uniqueidentifier NOT NULL,  
Owner nvarchar(255) NOT NULL,  
CustomSortOrder nvarchar(max) NULL,  
Name nvarchar(max) NOT NULL,  
Description nvarchar(1000) NULL,  
Type tinyint NOT NULL,  
IsOpen bit NOT NULL,  
AvailableForTagging bit NOT NULL,  
Stakeholders nvarchar(1000) NULL,  
Contact nvarchar(320) NULL,  
GroupId int NOT NULL,
```

**Id:** The internal identifier (1) of the term set.

**PartitionId:** The identifier of the term store partition to which the term set belongs.

**CreatedTime:** The UTC time when the term set was created.

**LastModifiedTime:** The UTC time of the last modification to the term set object.

**UniqueId:** The unique identifier of the term set

**Owner:** The name of the owner of the term set.

**CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the UniqueId column of the **ECMTerm** table (section [2.2.5.5](#)).

**Name:** The name of the term set. The value is a sequence of LCID|name separated by a semicolon. The name value in the sequence MUST NOT exceed 255 characters, and it MUST NOT contain any of the following characters:

- Semicolon (;)
- Double quotes (")
- Left angle bracket (<)
- Right angle bracket (>)
- Vertical bar (|)
- Ampersand (&)
- \t

**Description:** The description of the term set.

**Type:** The type of the term set. MUST be one of the following:

**IsOpen:** Specifies whether a term set allows users without explicit rights to the group to add terms to the term set.

**AvailableForTagging:** Specifies whether a term set should be visible to users that are not group administrators or term store administrators.

**Stakeholders:** A list of users that have an interest in the term set.

**Contact:** The user or group that should be contacted if a user wants to request the addition of a term to the term set, and the **IsOpen** flag of the term set is **false**.

**GroupId:** The internal identifier (1) of the group to which the term set belongs.

#### 2.2.5.4 ECMPackage

This table is used to store all the published and unpublished **content type packages** and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
Id varchar(512) NOT NULL,  
Type uniqueidentifier NOT NULL,  
IsPublished bit NOT NULL,  
LastModifiedTime datetime NOT NULL,
```

```
ContentSize int NOT NULL,  
Content varbinary(max) NULL,
```

**PartitionId:** The identifier of the partition that the content type belongs to.

**Id:** The identifier of the content type.

**Type:** The identifier of the type of object in the package. The value MUST be "B4AD3A44-D934-4C91-8D1F-463ACEADE443".

**IsPublished:** Specifies whether the content type is published.

**LastModifiedTime:** The UTC time of the last modification to the content type.

**ContentSize:** The size of the content type package content in bytes.

**Content:** Specifies information and content about the specified content type package. The first 16 bytes specifies the **PartitionId**. The bytes after the sixteenth byte and before the first encountered white space byte specify the identifier of a content type. The next 16 bytes MUST specify the GUID of "B4AD3A44-D934-4c91-8D1F-463ACEADE443". The rest of bytes specify the content of a content type package.

### 2.2.5.5 ECMTerm

This table is used to store all of the terms in the term store and is defined as follows.

```
Id int NOT NULL,  
PartitionId uniqueidentifier NOT NULL,  
CreatedTime datetime NOT NULL,  
LastModifiedTime datetime NULL,  
UniqueId uniqueidentifier NOT NULL,  
LastUsedTime datetime NULL,  
UseCount int NOT NULL,  
Owner nvarchar(255) NOT NULL,  
IsDeprecated bit NOT NULL,  
IsDeleted bit NOT NULL,  
MergedIdList varchar(max) NULL,
```

**Id:** The internal identifier (1) of the term.

**PartitionId:** The identifier of the term store partition to which the term belongs.

**CreatedTime:** The UTC time when the term was created.

**LastModifiedTime:** The UTC time of the last modification to the term object.

**UniqueId:** The unique identifier of the term.

**LastUsedTime:** Reserved. This value can be anything, but it is not used by the system and MUST be ignored.

**UseCount:** Reserved. This value can be anything, but is not used by the system and MUST be ignored.

**Owner:** The name of the owner of the term.

**IsDeprecated:** Specifies whether the term has been deprecated.

**IsDeleted:** Specifies whether the term has been deleted.

**MergedIdList:** A list of term unique identifiers separated by a backslash (\) that represents all of the terms that have been merged into the current term.

### 2.2.5.6 ECMTermLabel

This table is used to store all of the labels for the terms in the term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
TermId int NOT NULL,  
LCID int NOT NULL,  
Label nvarchar(255) NOT NULL,  
IsDefault bit NOT NULL,
```

**PartitionId:** The identifier of the term store partition to which the label's term belongs.

**TermId:** The internal identifier (1) of the term to which the label belongs.

**LCID:** The identifier of the **locale** that the label is for.

**Label:** The text of the label.

**IsDefault:** Specifies whether this label is the default label for the given LCID. A given term can only have one default label per LCID.

### 2.2.5.7 ECMLanguage

This table is used to store all the languages supported by the term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
WorkingLanguageId int NOT NULL,  
IsDefaultLanguage bit NOT NULL,
```

**PartitionId:** The identifier of the term store partition to which the language belongs.

**WorkingLanguageId:** The identifier of the corresponding locale for the language.

**IsDefaultLanguage:** Specifies whether the language is the default language for the term store partition. One term store partition can only have one language as the default language.

### 2.2.5.8 ECMUsedTerms

The table is used to store all the terms that are being used for tagging in site collections and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
WebAppId uniqueidentifier NOT NULL,  
ContentDatabaseId uniqueidentifier NOT NULL,  
SiteGuid uniqueidentifier NOT NULL,  
TermSetId uniqueidentifier NOT NULL,  
TermId int NOT NULL,
```

**PartitionId:** The identifier of the term store partition to which the term belongs.

**WebAppId:** The identifier of the **Web application** that contains the site collection in which the term is being used.

**ContentDatabaseId:** The identifier of the **content database** for the site collection in which the term is being used.

**SiteGuid:** The identifier of the site collection in which the term is being used.

**TermSetId:** The unique identifier of the term set to which the term belongs.

**TermId:** The internal identifier (1) of the term.

### 2.2.5.9 ECMergedTerm

This table is used to store all the terms that have been merged into other terms in the term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
TermUniqueId uniqueidentifier NOT NULL,  
TargetTermId int NOT NULL,
```

**PartitionId:** The identifier of the term store partition to which the merged term belongs.

**TermUniqueId:** The unique identifier of the merged term.

**TargetTermId:** The internal identifier (1) of the target term into which the term has been merged.

### 2.2.5.10 ECMTermProperty

This table is used to store all the properties of the terms in the term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
TermId int NOT NULL,  
TermSetId int NOT NULL,  
PropertyName nvarchar(255) NOT NULL,  
PropertyValue nvarchar(max) NOT NULL,
```

**PartitionId:** The unique identifier of the term store partition to which the term belongs.

**TermId:** The internal identifier (1) of the term to which the property belongs.

**TermSetId:** The internal identifier (1) of the term set of the term to which the property belongs. [<6>](#)

**PropertyName:** The name of the term property. The value MUST NOT exceed 255 characters, and it MUST NOT contain the following characters

- Semicolon (;)
- Double quotes (")
- Left angle bracket (<)
- Right angle bracket (>)

- Vertical bar (|)
- Ampersand (&)
- \t

**PropertyValue:** The value of the term property.

### 2.2.5.11 ECMGroup

This table is used to store all the groups in the term store and is defined as follows.

```

Id int NOT NULL,
PartitionId uniqueidentifier NOT NULL,
UniqueId uniqueidentifier NOT NULL,
Name nvarchar(255) NOT NULL,
Description nvarchar(1000) NULL,
LastModifiedTime datetime NULL,
CreatedTime datetime NOT NULL,
Type int NOT NULL,

```

**Id:** The identifier of the group.

**PartitionId:** The unique identifier of the term store partition that the group belongs to.

**UniqueId:** The unique identifier of the group.

**Name:** The name of the group. The value MUST NOT exceed 255 characters, and it MUST NOT contain the following characters

- Semicolon (;)
- Double quotes (")
- Left angle bracket (<)
- Right angle bracket (>)
- Vertical bar (|)
- Ampersand (&)
- \t

**Description:** The description of the group.

**LastModifiedTime:** The UTC time of last modification to the group.

**CreatedTime:** The UTC time when the group was created.

**Type:** The type of group.

### 2.2.5.12 ECMServiceSettings

This table is used to track settings per partition. It stores the partitions term store identifier as well as content type syndication settings and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
Settings nvarchar(max) NULL,
```

**PartitionId:** The unique identifier of the partition to which the settings belong.

**Settings:** Xml that specifies the term store identifier, the content type syndication hub URL, and whether content type syndication error reporting is enabled or not. The XML MUST conform to the [\[MS-EMMWCF\] ServiceSettingsSettingsXmlDoc](#) complex type.

### 2.2.5.13 ECMTermDescription

This table is used to store all of the descriptions for terms in the term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
TermId int NOT NULL,  
LCID int NOT NULL,  
Description nvarchar(1000) NOT NULL,
```

**PartitionId:** The unique identifier of the term store partition to which the term belongs.

**TermId:** The internal identifier (1) of the term that the description is for.

**LCID:** The identifier of the locale that the description is for.

**Description:** The description of the term.

### 2.2.5.14 ECMTermSetMembership

This table is used to store the term set hierarchy for term sets in the system. A term can only appear once in a given term set and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
TermSetId int NOT NULL,  
TermId int NOT NULL,  
Path varchar(884) NOT NULL,  
ParentTermId int NULL,  
AvailableForTagging bit NOT NULL,  
CustomSortOrder nvarchar(max) NULL,  
IsSource bit NOT NULL,  
PinSourceTermSetId int NULL,
```

**PartitionId:** The identifier of the term store partition to which the term hierarchy belongs.

**TermSetId:** The internal identifier (1) of the term set that the term membership is in.

**TermId:** The internal identifier (1) of the child term in the term set.

**Path:** Specifies the hierarchy (path) of the term in the term set. The value is composed of a sequence of term internal identifiers (1) separated by a backslash (\). The first internal identifier (1) is for the ancestor term at the top of the hierarchy. The last internal identifier (1) is for the current term. If the term is at the top of the hierarchy, the value is the internal identifier (1) of the term.

**ParentTermId:** The internal identifier (1) of the parent term of the current term in the term set. If the term is at the root of the term set, this will be zero.

**AvailableForTagging:** Specifies if the term is available for tagging when used in the context of the given term set.

**CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the UniqueId column of the **ECMTerm** table (section [2.2.5.5](#)).

**IsSource:** For terms that are reused in multiple term sets, one instance is designated as the source term, whose permissions are used for operations that affect all instances of the term. If the term has only one instance, then that instance is considered the source term.

**PinSourceTermSetId:** The internal identifier (1) of the source term set that this term is pinned to. If the term is not pinned to any term set, this MUST be NULL. <7>

### 2.2.5.15 ECMPermission

The table is used to store all of the group permissions in a term store and is defined as follows.

```
PartitionId uniqueidentifier NOT NULL,  
GroupId int NOT NULL,  
PrincipalName nvarchar(255) NOT NULL,  
DisplayName nvarchar(255) NOT NULL,  
RawSid varbinary(512) NULL,  
Rights bigint NOT NULL,
```

**PartitionId:** The identifier of the term store partition to which the group belongs.

**GroupId:** The internal identifier (1) of a group.

**PrincipalName:** The user NT account name, in Domain\Account format.

**DisplayName:** The users display name.

**RawSid:** Reserved. The value must be NULL.

**Rights:** Rights. The value is a bitmask that specifies rights of the user.

Possible parameter values:

Value	Description
0x000	No permissions.
0x001	Edit term.
0x002	Edit term set.
0x004	Edit group.
0x008	Add 'edit term set' permissions.
0x010	Manage term store.

Value	Description
0x020	Add 'manage term store' permissions.
0x040	Contributor.
0x080	Group manager.
0x100	Term store administrator.
0xFF	All permissions.

## 2.2.6 XML Structures

No common XML structures are defined in this protocol.

### 2.2.6.1 Namespaces

This specification defines and references various **XML namespaces** using the mechanisms specified in [\[XMLNS\]](#). Although this specification associates a specific namespace prefix for each namespace that is used, the choice of any particular namespace prefix is implementation-specific and not significant for interoperability.

### 2.2.6.2 Simple Types

This specification does not define any common XML Schema simple type definitions.

### 2.2.6.3 Complex Types

This specification does not define any common XML Schema complex type definitions.

### 2.2.6.4 Elements

This specification does not define any common XML Schema element definitions.

### 2.2.6.5 Attributes

This specification does not define any common XML Schema attribute definitions.

### 2.2.6.6 Groups

This specification does not define any common XML Schema group definitions.

### 2.2.6.7 Attribute Groups

This specification does not define any common XML Schema attribute group definitions.

## 2.2.7 Raised Errors

The following table describes error codes raised by these stored procedures.

ErrorID	Error text	Description
2	ECMTaxonomyError_2. Lcid: %d	This error is raised when stored procedure is unable to complete because of an LCID

ErrorID	Error text	Description
		parameter which is not one of the registered working languages. <b>%d:</b> The invalid working language LCID.
3	ECMTaxonomyError_3. Lcid: %d	This error is raised when stored procedure is unable to complete because working language is default for the term store partition. <b>%d:</b> The working language LCID.
4	ECMTaxonomyError_4.GroupInternalId: %d	This error is raised when stored procedure is unable to complete because of term group internal identifier (1) passed as parameter <b>%d:</b> The invalid term group internal identifier (1).
5	ECMTaxonomyError_5.GroupInternalId: %d	This error is raised when the term set group update failed because of a save conflict. <b>%d:</b> internal identifier (1) of the term group.
6	ECMTaxonomyError_6.GroupInternalId: %d	This error is raised when stored procedure is unable to complete because of term group internal identifier (1) passed as parameter. <b>%d:</b> The invalid term group internal identifier (1).
7	ECMTaxonomyError_7.GroupInternalId: %d	This error is raised when the term group deletion failed because of a save conflict. <b>%d:</b> The internal identifier (1) of the term group.
8	ECMTaxonomyError_8.TermSetInternalId: %d	This error is raised when stored procedure is unable to complete because of the term set internal identifier (1) passed as parameter. <b>%d:</b> The invalid internal identifier (1) of the term set.
9	ECMTaxonomyError_9.TermSetInternalId: %d	This error is raised when the term set update failed because of a save conflict. <b>%d:</b> The internal identifier (1) of the term set.
10	ECMTaxonomyError_10. TermInternalId: %d	This error is raised when stored procedure is unable to complete because of the term internal identifier (1) passed as parameter. <b>%d:</b> The invalid internal identifier (1) of the term.
11	ECMTaxonomyError_11. TermInternalId: %d	This error is raised when the stored procedure encounters a save conflict for a term. <b>%d:</b> The internal identifier (1) of the term.
12	ECMTaxonomyError_12. TermInternalId: %d.	This error is raised when stored procedure is unable to complete because term was reused in other term sets. <b>%d:</b> The internal identifier (1) of the term, internal identifier (1) of the parent term set of the term.

ErrorID	Error text	Description
13	ECMTaxonomyError_13.TermSetInternalId: %d	This error is raised when the term set deletion failed because of a save conflict. <b>%d:</b> The internal identifier (1) of the term set.
14	ECMTaxonomyError_14.TermSetInternalId: %d	This error is raised when stored procedure is unable to complete because one or more terms in the term set was reused in other term sets. <b>%d:</b> The internal identifier (1) of the term set.
15	ECMTaxonomyError_15. GroupInternalId: %d	This error is raised when stored procedure is unable to complete because of term group internal identifier (1) passed as parameter <b>%d:</b> The invalid term group internal identifier (1).
17	ECMTaxonomyError_17.	This error is raised when the stored procedure execution would result in multiple keywords with the same default term label in the keyword term set
18	ECMTaxonomyError_18.TermSetInternalId: %d	This error is raised when stored procedure is executed by users with insufficient permissions to add a term to a term set with its <b>IsOpen</b> property defined <b>ECMTermSet</b> set to <b>false</b> . <b>%d:</b> The internal identifier (1) of the term set.
19	ECMTaxonomyError_19.	This error is raised when stored procedure execution would result in multiple terms with same default term label and parent term.
20	ECMTaxonomyError_20.	This error is raised when the stored procedure is unable to complete because a term label already exists in the specified language for the term.
21	ECMTaxonomyError_21. %s	This error is raised when the stored procedure execution would result in multiple term sets with the same name in the same language. <b>%s:</b> The name of the term set being added or updated.
22	ECMTaxonomyError_22. %s	This error is raised when stored procedure execution would result in term sets sharing names in the same language. <b>%s:</b> The list of term sets having the same name.
23	ECMTaxonomyError_23. %s	This error is raised when stored procedure execution would result in terms having the same default label and parent terms. <b>%s:</b> The list of term sets having the same default label and parent terms.
24	ECMTaxonomyError_24. GroupInternalId: %d ECMTaxonomyError_24. TermSetInternalId: %d	This error is raised when the current user has insufficient permissions to perform this operation. <b>%d:</b> The internal identifier (1) of the term or

<b>ErrorID</b>	<b>Error text</b>	<b>Description</b>
	%d ECMTaxonomyError_24. TermInternalId: %d ECMTaxonomyError_24. ECMTaxonomyError_24. TermInternalId %d	the term set or the group.

Preliminary

## 3 Protocol Details

The client side of this protocol is simply a pass-through. That is, no additional timers or other state is required on the client side of this protocol. Calls made by the higher-layer protocol or application are passed directly to the transport, and the results returned by the transport are passed directly back to the higher-layer protocol or application.

This protocol allows protocol servers to notify protocol clients of stored procedure faults by raising errors defined in section [2.2.7](#).

### 3.1 Server Details

#### 3.1.1 Abstract Data Model

This section describes a conceptual model of a term store. The described model 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 purposes of clarity and illustration, each protocol server (or "service application") can be broken down into two parts. The term store which pertains to the management of hierarchical metadata and the content type publishing service which allows for replicating content types across site collections, Web applications, and Web farms.

##### Term store

A term store is unique to a **service application** and is made of groups which in turn contain term sets. A service application may contain multiple term stores.

Each term store maintains settings for:

- Language(s), including default language permissions.

Each term store also has settings that control whether it is partitioned or unpartitioned. That is, the data maintained by the protocol server can either be stored for all callers or partitioned by a **partition identifier**. Unpartitioned term stores are equivalent to a single partition.

Each group allows for different ACLs to be set to control the users who have permission to edit and manage the term sets. Terms sets are collections of terms, each one consisting of one or more labels and descriptions. Many of the common action a user can do to a term are:

- Merge-Delete-Deprecate-Translate-Move

Term sets contain a hierarchy of terms (potentially re-using individual terms in multiple term sets, for example "Contoso" being reused as a vendor, a partner and employer); that is, the term set has root terms (terms without parents), and each term has zero or more child terms.

Each term set and each term has values for one or more of the languages. For example, a term has a default term label for each language, and may have multiple term labels besides the default term label for each language. A keyword only has one term label.

For terms that are reused in multiple term sets, one instance is designated as the source term, whose permissions are used for operations that affect all instances of the term. If the term has only one instance, then that instance is considered the source term.

## Content type publishing

The protocol server, or service application, also maintains for each partition the following settings:

- The location of the content type publication hub, that is, the URL of a site collection that contains content types that are shared for usage.
- Content type packages that contain the definitions of those content types.
- A log for issues encountered during content type publication, and a setting controlling whether errors are logged or not.

The content type publishing functionality allows the service application to specify a location, the hub, from which to push content types to other site collections that have opted to receive the content types. This allows an enterprise to centrally manage and control the schema for content items across all participating site collections, regardless of farm or site collection boundaries.

### 3.1.2 Timers

None.

### 3.1.3 Initialization

Before any other method can be called on a term set, the **GetSessionData** method MUST be called.

### 3.1.4 Higher-Layer Triggered Events

None.

### 3.1.5 Message Processing Events and Sequencing Rules

Transactions containing any of the stored procedures that are part of an add/update/delete operation MUST be executed in the order specified by these steps:

1. Begin transaction.
2. Execute the following batch of SQL instructions:

```
IF OBJECT_ID('tempdb..#GroupIds') IS NOT NULL
    DROP TABLE #GroupIds
CREATE TABLE #GroupIds(OldId int PRIMARY KEY, NewId int)

IF OBJECT_ID('tempdb..#TermSetIds') IS NOT NULL
    DROP TABLE #TermSetIds
CREATE TABLE #TermSetIds(OldId int PRIMARY KEY, NewId int)

IF OBJECT_ID('tempdb..#TermIds') IS NOT NULL
    DROP TABLE #TermIds
CREATE TABLE #TermIds(OldId int PRIMARY KEY, NewId int)

IF OBJECT_ID('tempdb..#ChangeCache') IS NOT NULL
    DROP TABLE #ChangeCache
CREATE TABLE #ChangeCache
(PartitionId uniqueidentifier NOT NULL
,GroupUniqueId uniqueidentifier
,TermSetUniqueId uniqueidentifier
,ObjectUniqueId uniqueidentifier)
```

```

,ObjectId int
,ObjectType int NOT NULL
,ChangeType int NOT NULL
,ChangeTime datetime NOT NULL
,ChangeData nvarchar(max) NULL
,ModifiedBy nvarchar(255) NULL)

```

3. Call the desired stored procedure(s).
4. Execute **proc\_ECM\_LogChange**.
5. Commit or rollback transaction.

### 3.1.5.1 proc\_ECM\_AddTermLabel

This stored procedure is called to add a term label. It creates an entry in the table described in section [2.2.5.6](#). Adding a new default term label overrides any pre-existing default term labels in that language for the term. The Initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure MUST be executed after **proc\_ECM\_CreateTerm** or **proc\_ECM\_UpdateTerm** is executed. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_AddTermLabel (
  @PartitionId uniqueidentifier
  ,@TermId int
  ,@LCID int
  ,@Label nvarchar(255)
  ,@IsDefault bit = 0
  ,@IsKeyword bit = 0
  ,@IsFirstOnly bit = 0
);

```

**@PartitionId:** term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term for which the term label is being added.

**@LCID:** The LCID that indicates the language of the term label being added. Protocol client MUST provide a valid LCID from the **WorkingLanguageId** column of the **ECMLanguage** table (section [2.2.5.7](#)).

**@Label:** Term label. Value of this parameter MUST NOT exceed 255 characters. The value of this parameter MUST NOT contain any of the following characters:

- Semicolon (;)
- Double quotes (")
- Left angle bracket (<)
- Right angle bracket (>)
- Vertical bar (|)
- Ampersand (&)

Term labels in a particular language MUST be unique for a term.

**@IsDefault:** Indicates whether the term label being added is the default term label for the term in the language specified by the **@LCID** parameter. If the value of this parameter is 1 and a sibling term with the same default term label in the same language already exists then error **ECMTaxonomyError\_19** described in section [2.2.7](#) MUST be raised.

**@IsKeyword:** Indicates whether the term label belongs to the keyword term set of the term store. If the value of this parameter is 1 and a keyword with the same default term label already exists in the keywords term set then error **ECMTaxonomyError\_17** described in section [2.2.7](#) MUST be raised.

**@IsFirstOnly:** Indicates whether the user only has restricted write permissions. If the value of this parameter is 1, this operation is restricted to adding terms with single term label to open term sets in a non-site collection group, or adding term labels to term sets in site collection groups. If the term is in a closed term sets in a non-site collection group, error **ECMTaxonomyError\_24** described in section [2.2.7](#) MUST be thrown. If value is 1 and a term label already exists for this term in this language, error **ECMTaxonomyError\_20** described in section [2.2.7](#) MUST be raised.

**Return Values:** 0 for success, or a value that MUST be in the following table.

Value	Description
<b>ECMTermLabel.@@error</b>	Type the return value description.
<b>-1</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.2 **proc\_ECM\_AddApplicationLogEntry**

This stored procedure is called to add a log entry for content type syndication. It creates an entry in the table described in section [2.2.5.1](#). This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddApplicationLogEntry (  
    @PartitionId uniqueidentifier  
    ,@SiteUrl nvarchar(2000)  
    ,@ObjectName nvarchar(max)  
    ,@Stage varchar(100)  
    ,@Message nvarchar(max)  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@SiteUrl:** The URL of the site in which the log entry is generated. This parameter MUST NOT be NULL.

**@ObjectName:** The name of the object on which the log entry is generated.

**@Stage:** The name of the content type syndication stage at which the log entry is generated. This parameter MUST NOT be NULL.

**@Message:** The content of the log entry. This parameter MUST NOT be null.

**Return Values:** An integer which MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.3 proc\_ECM\_AddLanguage

The **proc\_ECM\_AddLanguage** stored procedure is called to add a new language to the list of the languages supported by a term store. On success, and after **proc\_ECM\_LogChange** is called and the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeAdd** and an **ObjectType** of **ItemTypeTermStore**. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddLanguage (  
    @PartitionId uniqueidentifier  
    ,@WorkingLanguageId int  
    ,@IsDefaultLanguage bit = 0  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@WorkingLanguageId:** The LCID that identifies the language being added. This parameter MUST NOT be NULL. When the list of the languages supported by the term store is empty, **proc\_ECM\_AddLanguage** MUST mark the language being added as the default for the term store.

**@IsDefaultLanguage:** A numeric value that indicates whether the language being added MUST be set as the default language for the term store. This parameter MUST NOT be NULL. When **@IsDefaultLanguage** is set to 1 and the term store previously had a different default language, **proc\_ECM\_AddLanguage** MUST mark the previous default language as non-default; it MUST also add a new entry in the **ECMChangeLog** table with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeTermStore**. When **@IsDefaultLanguage** is set to 1 and there are terms which do not have default term labels for the language being added, **proc\_ECM\_AddLanguage** MUST copy over the default label for the previous default language for all such terms.

**Return Values:** An integer which MUST be in the following table.

Value	Description
<b>ECMLanguage.@@error</b>	Type the return value description.
<b>0</b>	Default return value
<b>-1</b>	Returned if <b>@IsDefaultLanguage = 1</b> and the stored procedure failed to populate names of existing termsets and the corresponding default labels for the new language that was specified to be added.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.4 proc\_ECM\_AddTermDescription

This stored procedure is called to add a description for the term. It creates an entry in the table described in section [2.2.5.13](#). The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddTermDescription (  
    @PartitionId uniqueidentifier  
    ,@TermId int  
    ,@LCID int  
    ,@Description nvarchar(1000)
```

);

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term for which description is being added.

**@LCID:** The LCID that identifies the language of the description. Protocol client MUST provide a valid LCID from the **WorkingLanguageId** column of the **ECMLanguage** table (section [2.2.5.7](#)).

**@Description:** Description of the term. The description for a term helps to understand the intended use of a term and is also useful for disambiguation. Length of the description MUST NOT exceed 1000 characters.

**Return Values:** 0 for success, or an integer that MUST be in the following table.

Value	Description
@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.5 proc\_ECM\_AddTermProperty

This stored procedure is called to add a custom term property. It creates an entry in the table described in section [2.2.5.10](#). The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddTermProperty (  
    @PartitionId uniqueidentifier  
    ,@TermId int  
    ,@TermSetId int  
    ,@PropertyName nvarchar(255)  
    ,@PropertyValue nvarchar(max)  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term for which a custom property is being added.

**@TermSetId:** Internal identifier (1) of the term set to which the term belongs. [<8>](#)

**@PropertyName:** Name of the custom term property being added.

**@PropertyValue:** Value of the custom term property being added

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.6 proc\_ECM\_AddTermSetMembership

This stored procedure adds the specified term from the specified term store partition as a member of the specified term set. On success, and after `proc_ECM_LogChange` is called and the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeAdd** and an **ObjectType** of **ItemTypeTerm**. The Initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. If adding the specified term as a member of the specified term set would result in multiple terms with the same default label and parent term, then the error **ECMTaxonomyError\_19** described in error section [2.2.7](#) MUST be raised. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddTermSetMembership (  
  @PartitionId uniqueidentifier  
  ,@TermSetId int  
  ,@TermId int  
  ,@ParentTermId int = 0  
  ,@AvailableForTagging bit = 0  
  ,@CustomSortOrder nvarchar(max) = null  
  ,@IsSource bit  
  ,@SourceId int = null  
  ,@IsRestrictedOpen bit  
  ,@lastModifiedTime datetime = null  
  ,@PinSourceTermSetId int = null  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermSetId:** Internal identifier (1) of the term set to which a term is being added as a member. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term being added. This parameter MUST NOT be NULL.

**@ParentTermId:** Internal identifier (1) of the parent term of the term being added. This parameter MUST NOT be NULL. If the specified parent term is "0", meaning that the term is a root term, **LastModifiedTime** value in the **ECMTermSet** table (section [2.2.5.3](#)) row corresponding to the specified term set MUST be updated. Otherwise, **LastModifiedTime** value in the **ECMTerm** table (section [2.2.5.5](#)) row corresponding to the specified parent term MUST be updated. **LastModifiedTime** must be updated with the **@lastModifiedTime** value when that parameter is NOT null; otherwise the current time in UTC.

**@AvailableForTagging:** Indicates whether the term being added is available to be used by end users for tagging. This parameter MUST NOT be NULL.

**@CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form `identifier1:identifier2...` where `identifiern` is a valid GUID from the **UniqueId** column of the **ECMTerm** table.

**@IsSource:** Indicates whether the term being added to the term set is a source term. This parameter MUST NOT be NULL. When set to 1, any instance of the specified term belonging to the term set other than the specified one MUST be set as NOT a source term.

**@SourceId:** The internal identifier (1) of another term that the term being added was copied from, if any.

**@IsRestrictedOpen:** Indicates whether a term MUST only be added as a member of an **open term set**. An open term set is indicated by the **IsOpen** column of the **ECMTermSet** table. When set to 1, and the specified term set is NOT an open term set, the error **ECMTaxonomyError\_18** described in error section [2.2.7](#) MUST be raised and return value of -1 returned.

**@lastModifiedTime:** The last modified time of the term being added in UTC format or NULL.

**@PinSourceTermSetId:** The internal identifier (1) of the source term set that this term is pinned to. If the term is not pinned to any term set, this MUST be NULL. <9>

**Return Values:** 0 for success or an integer which MUST be in the following table.

Value	Description
-1	Adding term as a member of the term set failed because the error <b>ECMTaxonomyError_18</b> described in section <a href="#">2.2.7</a> was raised.

### Result Sets:

This stored procedure MUST return a [proc ECM MoveTermSetMembership.ResultSet0](#)

### 3.1.5.7 proc\_ECM\_AddUsedTerm

This stored procedure is called to add a term from a term store partition to the list of terms used by a site collection. On success, and after the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeAdd** and an **ObjectType** of **ItemTypeSite**. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddUsedTerm (  
  @PartitionId uniqueidentifier  
  ,@WebAppId uniqueidentifier  
  ,@ContentDatabaseId uniqueidentifier  
  ,@SiteGuid uniqueidentifier  
  ,@TermSetId uniqueidentifier  
  ,@TermId uniqueidentifier  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@WebAppId:** The identifier of the Web application which contains the site collection in which the identified term is being used. This parameter MUST NOT be NULL.

**@ContentDatabaseId:** The identifier of the content database for the site collection in which the identified term is being used. This parameter MUST NOT be NULL.

**@SiteGuid:** The identifier of the site collection in which the identified term is being used. This parameter MUST NOT be NULL.

**@TermSetId:** The identifier of the term set to which the identified term belongs. This parameter MUST NOT be NULL.

**@TermId:** Term identifier. This parameter MUST NOT be NULL.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
@@error	An unexpected error occurred.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.8 proc\_ECM\_ClearPartitionData

This stored procedure is called to delete all data for a particular term store partition in the following tables:

- **ECMUsedTerms** (section [2.2.5.8](#))
- **ECMTermSetMembership** (section [2.2.5.14](#))
- **ECMTermLabel** (section [2.2.5.6](#))
- **ECMTermDescription** (section [2.2.5.13](#))
- **ECMTermProperty** (section [2.2.5.10](#))
- **ECMPermission** section [2.2.5.15](#)()
- **ECMChangeLog** (section [2.2.5.2](#))
- **ECMTermSet** (section [2.2.5.3](#))
- **ECMMergedTerm** (section [2.2.5.9](#))
- **ECMTerm** (section [2.2.5.5](#))
- **ECMGroup** (section [2.2.5.11](#))
- **ECMLanguage** (section [2.2.5.7](#))
- **ECMServiceSettings** (section [2.2.5.12](#))
- **ECMPackage** (section [2.2.5.4](#))
- **ECMApplicationLog** (section [2.2.5.1](#))

After this stored procedure completes successfully, the log of changes MUST contain an entry with **PartitionId** as **@PartitionId**, **ObjectType** as **ItemTypeTermStore**, **ChangeType** as **OperationTypeDelete**, **ChangeTime** as current UTC time, and the rest as NULL to note the specified term store partition was deleted. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_ClearPartitionData (
  @PartitionId uniqueidentifier
);
```

**@PartitionId:** Identifier of the term store partition for which all data is to be deleted. This parameter MUST NOT be NULL.

**Return Values:** An integer which MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.9 proc\_ECM\_CreateGroup

This stored procedure creates a term set group in the partition. It creates an entry in the table **ECMGroup** (section [2.2.5.11](#)). On success, and after [proc\\_ECM\\_LogChange](#) is called and the transaction is committed, there MUST be a new entry in the [ECMChangeLog](#) table with a **ChangeType** of **OperationTypeAdd** and an **ObjectType** of **ItemTypeGroup**. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_CreateGroup (  
  @PartitionId uniqueidentifier  
  ,@UniqueId uniqueidentifier  
  ,@Name nvarchar(255)  
  ,@Description nvarchar(1000) = null  
  ,@Type int = 0  
  ,@OldId int  
);
```

**@PartitionId:** Partition identifier of term group to be added.

**@UniqueId:** term group identifier of term group to be added.

**@Name:** Name of term group to be added.

**@Description:** Description of term group to be added.

**@Type:** Type of term group. MUST be either 0 (regular term group), 1 (system term group), or 2 (site collection term group).

**@OldId:** A temporary internal identifier (1) for the term group. Value MUST be a negative integer. This identifier is used to link operations to a term group if a batch of operations is being executed on the new term group.

**Return Values:** An integer which MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.10 proc\_ECM\_CreatePermission

This stored procedure is called to create an ACL for the term store or term set group. It creates an entry in the table **ECMPermission** (section [2.2.5.15](#)). The ACL assigns the user one or more of the following rights: none, edit term, edit term set, edit term group, add term set, edit permissions, manage term store, contributor, term group manager, term store administrator. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_CreatePermission (  
  @PartitionId uniqueidentifier  
  ,@GroupId int  
  ,@PrincipalName nvarchar(255)  
  ,@DisplayName nvarchar(255)  
  ,@Rights bigint  
  ,@AddToLog bit = 0  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@GroupId:** Internal identifier (1) of the term group. If nonzero and a corresponding term group does not exist then the error **ECMTaxonomyError\_15** described in error section [2.2.7](#) MUST be raised. If the value is 0, permissions are created to the term store.

**@PrincipalName:** The NT account, in Domain\Account format, of the user for whom permissions are being created.

**@DisplayName:** The full name of the user for whom permissions are being created.

**@Rights:** Rights. This defines the rights of the user to the term store or a term group. Value MUST NOT be NULL. Value MUST be one of the values defined for the Rights column in the **ECMPermission** table.

**@AddToLog:** Indicates whether this action is noted in the log of changes. If the value is 1 and the value of the **@GroupId** parameter is 0, after this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeTermStore**. If the value is 1 and the value of the **@GroupId** parameter is not 0 then after this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** table with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeGroup**.

**Return Values:** 0 for success or an integer that MUST be 0 or in the following table.

Value	Description
-1	Permission was not created because the error <b>ECMTaxonomyError_15</b> described in error section <a href="#">2.2.7</a> was raised.
@@error	Type the return value description.
ECMGroup.@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.11 proc\_ECM\_CreateTerm

This stored procedure creates an entry in the [ECMTerm](#) table. This is the first step toward creating a term. The stored procedure to add a term label MUST be called to complete the term creation operation. The stored procedure to add term set membership for the term is executed and the corresponding results also apply to this stored procedure. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_CreateTerm (
  @PartitionId uniqueidentifier
  ,@Owner nvarchar(255)
  ,@UniqueId uniqueidentifier
  ,@UseCount int = 0
  ,@IsDeprecated bit = 0
  ,@OldId int
  ,@TermSetId int
  ,@ParentTermId int = 0
  ,@AvailableForTagging bit = 0
  ,@CustomSortOrder nvarchar(max) = null
  ,@IsSource bit = 1
```

```

    ,@SourceId int = null
    ,@IsRestrictedOpen bit
    ,@PinSourceTermSetId int = null
);

```

**@PartitionId:** Term store partition identifier (1). This parameter MUST NOT be NULL.

**@Owner:** The **user name** for the user creating the term.

**@UniqueId:** GUID that uniquely identifies the term.

**@UseCount:** Reserved. MUST be 0.

**@IsDeprecated:** Indicates whether this term is a **deprecated term**.

**@OldId:** A temporary internal identifier (1) for the term. Value MUST be a negative integer. This identifier is used to link operations to a term if a batch of operations is being executed on the new term.

**@TermSetId:** Internal identifier (1) of the parent term set of the term being created.

**@ParentTermId:** Internal identifier (1) of the parent term of the term being created. If the term is being added to the root, the **ParentTermId** is 0. This value MUST NOT be NULL.

**@AvailableForTagging:** Indicates whether this term is available to be used by end users for tagging.

**@CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the **UniqueId** column of the **ECMTerm** table (section [2.2.5.5](#)).

**@IsSource:** Indicates whether the term being created is a source term.

**@SourceId:** The internal identifier (1) of another term that the term being added was copied from, if any.

**@IsRestrictedOpen:** Indicates whether a corresponding entry can be created in the **ECMTermSetMembership** table (section [2.2.5.14](#)) for a term set that is not open. Entry is created when the stored procedure **proc\_ECM\_AddTermSetMembership** (section [3.1.5.6](#)) is executed. An open term set is indicated by the **IsOpen** column of the **ECMTermSet** table (section [2.2.5.3](#)).

**@PinSourceTermSetId:** The internal identifier (1) of the source term set that this term is pinned to. If the term is not pinned to any term set, this MUST be NULL. [<10>](#)

**Return Values:** 0 for success.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.12 proc\_ECM\_CreateTermSet

This stored procedure creates a new term set. This new term set MUST not use the same name in any working language as any other existing term set in the same group; otherwise error **ECMTaxonomyError\_21** described in error section [2.2.7](#) MUST be thrown. After this stored procedure is called and the transaction is committed, if the value of the **@SourceId** parameter is null then there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a

**ChangeType** of **OperationTypeAdd** and an **ObjectType** of **ItemTypeTermSet**. Otherwise there MUST be a new entry in the ECMChangeLog table with a **ChangeType** of **OperationTypeCopy** and an **ObjectType** of **ItemTypeTermSet**, and all terms in the term set specified by the **@SourceId** parameter value become reused in the new term set. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_CreateTermSet (  
    @PartitionId uniqueidentifier  
    ,@UniqueId uniqueidentifier  
    ,@Name nvarchar(max)  
    ,@Description nvarchar(1000) = null  
    ,@Owner nvarchar(255)  
    ,@CustomSortOrder nvarchar(max) = null  
    ,@Type tinyint = 0  
    ,@IsOpen bit = 0  
    ,@AvailableForTagging bit = 0  
    ,@Stakeholders nvarchar(1000) = null  
    ,@Contact nvarchar(320) = null  
    ,@GroupId int  
    ,@OldId int  
    ,@SourceId int = null  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@UniqueId:** Identifier for the term set.

**@Name:** Names of the term set being created. The value is a sequence of LCID|name separated by semicolon. The name value in the sequence MUST NOT exceed 255 characters, and it MUST NOT contain any of the following characters:

- Semicolon (;)
- Double quotes (")
- Angle brackets (< >)
- Vertical bar (|)
- Ampersand (&)
- \t

**@Description:** Text describing the term set being created. Description of a term set helps to understand the intended use of the term set. Length of the description MUST NOT exceed 1000 characters.

**@Owner:** The user name of the user creating the term set.

**@CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the UniqueId column of the **ECMTerm** table (section [2.2.5.5](#)).

**@Type:** Term set type. MUST be 0 or 1.

**@IsOpen:** Indicates whether the term set being created is an open term set.

**@AvailableForTagging:** Indicates whether this term set is available to be used by end users for tagging.

**@Stakeholders:** Names of users in the organization that should be notified before any major changes are made to the term set. If not NULL, the names provided MUST be separated by a semicolon.

**@Contact:** E-mail address of the user to who term suggestion and feedback is to be provided.

**@GroupId:** Internal identifier (1) of the parent group of the term set.

**@OldId:** Temporary internal identifier (1). Value MUST be a negative integer. This identifier is used to link operations to a term set if a batch of operations is being executed on the new term set.

**@SourceId:** The internal identifier (1) of another term set that the term set being created was copied from, if any. If not NULL, protocol client MUST provide a valid identifier from the Id column of the ECMTermSet table.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
ECMTermSetMembership.@@error	Type the return value description.
-1	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.13 proc\_ECM\_DeleteGroup

This stored procedure is used to delete a term set group that has no term sets in it. When this stored procedure runs successfully the row in **ECMPermission** identified by **@PartitionId** and **@Id** MUST be deleted, as well as the row in **ECMGroup** identified by **@PartitionId** and **@Id** MUST be deleted. Also after **proc\_ECM\_LogChange** (section 3.1.5.39) is called and the transaction has been committed, an entry in **ECMChangeLog** MUST have been added where **PartitionId** is **@PartitionId**, **GroupUniqueId** is the term group identifier identified by **@Id**, **ObjectId** is the **@Id** internal identifier (1), **ObjectType** is 3 (**ItemTypeGroup**), **ChangeType** is 3 (**OperationTypeDelete**), **ChangeTime** is the current time, and **TermSetUniqueId**, **ChangeData** and **ModifiedBy** are NULL. The **Initialization** routine specified in step 2 of section 3.1.5 MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteGroup (  
    @PartitionId uniqueidentifier  
    ,@Id int  
    ,@LastModifiedTime datetime = null  
);
```

**@PartitionId:** Partition identifier of term group to be deleted.

**@Id:** Internal identifier (1) of term group to be deleted.

**@LastModifiedTime:** The last modified time of term group that is being updated in UTC format or NULL.

**Return Values:** 0 for success or an integer which MUST be in the following table.

Value	Description
-1	MUST be returned if the term group has any existing term sets that belong to it. Also MUST be returned if <b>@LastModifiedTime</b> is not NULL and is less than the existing row <b>ECMGroup.LastModifiedTime</b> .
<b>ECMGroup.@@error</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.14 **proc\_ECM\_DeleteLanguage**

The **proc\_ECM\_DeleteLanguage** stored procedure is called to remove a language from the list of languages supported by the term store. The **proc\_ECM\_DeleteLanguage** stored procedure MUST NOT remove the term labels corresponding to the specified language. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteLanguage (
  @PartitionId uniqueidentifier
  ,@WorkingLanguageId int
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@WorkingLanguageId:** The LCID that identifies the language being removed. This parameter MUST NOT be NULL. If the **ECMLanguage** table (section [2.2.5.7](#)) contains a row corresponding to the **@WorkingLanguageId** and **@PartitionId** parameters before the execution of **proc\_ECM\_DeleteLanguage** stored procedure, on success, and after **proc\_ECM\_LogChange** is called and the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeDelete** and an **ObjectType** of **ItemTypeTerm** store.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
-1	<b>ECMLanguage</b> table contains a row with <b>WorkingLanguageId</b> column value of <b>@WorkingLanguageId</b> and <b>IsDefaultLanguage</b> column value of 1.
<b>@@error</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.15 **proc\_ECM\_DeletePermission**

This stored procedure is called to delete the ACL associated with a user for the term store or term set group. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeletePermission (
  @PartitionId uniqueidentifier
  ,@GroupId int
```

```
,@PrincipalName nvarchar(255)
,@AddToLog bit = 0
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@GroupId:** Internal identifier (1) of the term group for which ACL is being deleted. If nonzero and a corresponding term group does not exist then the error **ECMTaxonomyError\_15** described in error section [2.2.7](#) MUST be raised.

**@PrincipalName:** The NT account, in Domain\Account format, of the user whose permissions to the term group specified by the **@GroupId** parameter is being deleted. If the **@GroupId** parameter is 0 then permissions to the term store is deleted.

**@AddToLog:** Indicates whether this action is noted in the log of changes. If the value is 1 and the value of the **@GroupId** parameter is 0 then after this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeTermStore**. If the value is 1 and the value of the **@GroupId** parameter is not 0 then after this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** table with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeGroup**.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
-1	Deletion failed because the error <b>ECMTaxonomyError_15</b> described in error section <a href="#">2.2.7</a> was raised.
@@error	Type the return value description.
ECMGroup.@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.16 proc\_ECM\_DeleteServiceSettings

This stored procedure is called to delete service settings data in the **ECMServiceSettings** table (section [2.2.5.12](#)) for a term store partition. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteServiceSettings (
@PartitionId uniqueidentifier
);
```

**@PartitionId:** Partition identifier of the settings that MUST be deleted.

**Return Values:** 0 for success or an integer which MUST be in the following table.

Value	Description
@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.17 proc\_ECM\_DeleteTerm

This stored procedure is called to delete a term. If the term being deleted is a source term then it MUST be the last copy in the term store otherwise the error described by [ECMTaxonomyError\\_12](#) in section [2.2.7](#) is raised. After this stored procedure is called and the transaction is committed there MUST be a new entry in the [ECMChangeLog](#) table with a **ChangeType** of **OperationTypeDelete** and an **ObjectType** of **ItemTypeTerm**. The Initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteTerm (  
  @PartitionId uniqueidentifier  
  ,@TermId int  
  ,@TermSetId int  
  );
```

**@PartitionId:** Term store partition identifier (1). This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term being deleted.

**@TermSetId:** Internal identifier (1) of the parent term set of the term being deleted.

**Return Values:** An integer that MUST be in the following table.

Value	Description
0	The delete was successful.
-1	Delete failed because there were references to the term in the term store and error <b>ECMTaxonomyError_12</b> described in section <a href="#">2.2.7</a> was raised.
@deletedterms.@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.18 proc\_ECM\_DeleteTermDescription

This stored procedure is called to delete the description of a term. The Initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteTermDescription (  
  @PartitionId uniqueidentifier  
  ,@TermId int  
  ,@LCID int  
  );
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term whose description is being deleted.

**@LCID:** The LCID that indicates the language of the description being deleted.

**Return Values:** An integer that MUST be zero.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.19 proc\_ECM\_DeleteTermLabel

This stored procedure is called to delete a term label. The initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteTermLabel (  
  @PartitionId uniqueidentifier  
  ,@TermId int  
  ,@LCID int  
  ,@Label nvarchar(255)  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term whose label is being deleted.

**@LCID:** The LCID that indicates the language of the term label being deleted.

**@Label:** Term label being deleted. This parameter is not case-sensitive.

**Return Values:** An integer that MUST be zero.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.20 proc\_ECM\_DeleteTermProperty

This stored procedure is called to delete a custom term property. The initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteTermProperty (  
  @PartitionId uniqueidentifier  
  ,@TermId int  
  ,@TermSetId int  
  ,@PropertyName nvarchar(255)  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** The internal identifier (1) of the term whose property is being deleted.

**@TermSetId:** The internal identifier (1) of the term set to which the term belongs. [<11>](#)

**@PropertyName:** Name of the custom term property being deleted.

**Return Values:** An integer that MUST be zero.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.21 proc\_ECM\_DeleteTermSet

This stored procedure is called to delete a term set and all terms within it. If the source terms in the term set being deleted are being reused in the term set group then the error described by **ECMTaxonomyError\_14** in section [2.2.7](#) MUST be raised. After this stored procedure is called and

the transaction is committed there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeDelete** and an **ObjectType** of **ItemTypeTermSet**. The initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DeleteTermSet (
  @PartitionId uniqueidentifier
  ,@Id int
  ,@LastModifiedTime datetime = null
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@Id:** Internal identifier (1) of the term set being deleted.

**@LastModifiedTime:** The date and time in UTC format that indicates whether the protocol client has the latest view of the term set being deleted. If the value is not NULL and is not a future time as compared to the **LastModifiedTime** column of the **ECMTermSet** table (section [2.2.5.3](#)) for the term set being deleted then the error **ECMTaxonomyError\_13** described in section [2.2.7](#) MUST be raised.

**Return Values:** 0 for success or an integer which MUST be in the following table.

Value	Description
-1	Deletion failed because of error <b>ECMTaxonomyError_13</b> described in section <a href="#">2.2.7</a> or error <b>ECMTaxonomyError_14</b> because one or more source terms in the term set were reused in other term sets in the term store.
<b>@deletedterms.@@error</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.22 proc\_ECM\_DoesPartitionDataExist

This stored procedure is called to find whether a term store partition, specified by the **@PartitionId** parameter, exists in the content database of the service application. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_DoesPartitionDataExist (
  @PartitionId uniqueidentifier
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [DoesPartitionDataExistResultSet](#)

### 3.1.5.23 proc\_ECM\_GetAllTerms

The **proc\_ECM\_GetAllTerms** stored procedure is called to retrieve all the terms in the specified term set and the specified term store partition. If the term store partition does not exist, or if the term set does not exist within the term store partition, no data is returned.

Each result set MUST contain records sorted according to their taxonomy hierarchy; that is, parent terms are returned before child terms. If the specified term set exists in the term store partition, but contains no terms, the result sets 0 through 5 MUST all be empty. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetAllTerms (  
    @PartitionId uniqueidentifier  
    ,@TermSetId uniqueidentifier  
);
```

**@PartitionId:** The identifier of the term store partition that the term set belongs to. This parameter MUST NOT be NULL.

**@TermSetId:** The identifier of the term set whose terms are to be retrieved. This parameter MUST NOT be NULL.

**Return Values:** An integer which MUST be 0.

#### Result Sets:

For the following combination of parameters:

- **@PartitionId:** 0c37852b-34d0-418e-91c6-2ac25af4be5b
- **@TermSetId:** 5FC99D73-97CD-4D98-9403-7F01EF1783E9

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

### 3.1.5.24 proc\_ECM\_GetChanges

The **proc\_ECM\_GetChanges** stored procedure is called to retrieve details of changes in a term store partition or all term store partitions occurring after a specified time. If the term store partition identifier is not NULL, the changes from the specified partition is retrieved. Otherwise, the changes from all partitions are retrieved.

The details of changes applicable to all term store partitions occurring after the specified time MUST also be retrieved. Details of changes MUST be ordered by the time the change occurred in ascending order. If term set identifier is not NULL, the details of changes in the specified term set MUST be retrieved. Otherwise, if group identifier is not NULL, the details of changes in the specified group MUST be retrieved. Otherwise, the details of changes retrieved MUST NOT be restricted to any

particular term set or group. Additionally, if term set identifier is not NULL or group identifier is not NULL, changes to default language of the specified term store partition MUST also be retrieved from records of changes where term set identifier or group identifier is NULL correspondingly. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetChanges (  
    @PartitionId uniqueidentifier = null  
    ,@GroupId int = null  
    ,@TermSetId int = null  
    ,@SinceTime datetime = null  
    ,@ChangedItemType int = null  
    ,@ChangedOperationType int = null  
);
```

**@PartitionId:** Term store partition identifier.

**@GroupId:** Group identifier on which to filtered.

**@TermSetId:** term set identifier on which to filtered.

**@SinceTime:** The date and time value, in UTC, after which the changes that occurred MUST be retrieved. This parameter MUST NOT be NULL.

**@ChangedItemType:** Type of item to which the change applies. When NULL, MUST have no effect on set returned. Otherwise, MUST retrieve only the details of changes associated with specified value. Non-NULL values MUST be as specified for column **ObjectType** of **ECMChangeLog** table (section [2.2.5.2](#)).

**@ChangedOperationType:** Type of operation that caused the change. When NULL, MUST have no effect on set returned. Otherwise, MUST retrieve only the details of changes associated with specified value. Non-NULL values MUST be as specified for column **ChangeType** of **ECMChangeLog** table.

**Return Values:** An integer that MUST be 0.

#### **Result Sets:**

For the following combination of parameters:

- **@PartitionId:** a valid partition identifier.
- **@SinceTime:** 1/1/1.

This stored procedure MUST return a [proc\\_ECM\\_GetChanges.proc\\_ECM\\_GetChanges.Default.ResultSet0](#)

### **3.1.5.25 proc\_ECM\_GetChangeTime**

The **proc\_ECM\_GetChangeTime** stored procedure is called to get the latest time anything within a term store partition, group or term set has changed. If term set identifier is not NULL, the latest time anything within the specified term set within the specified term store partition or affecting all partitions was changed is retrieved. Otherwise, if group identifier is not NULL, the latest time anything within the specified group within the specified term store partition or affecting all partitions was changed is retrieved. Otherwise, the latest time anything within the specified term store partition or affecting all partitions was changed is retrieved. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_GetChangeTime (
  @PartitionId uniqueidentifier
  ,@GroupId uniqueidentifier = null
  ,@TermSetId uniqueidentifier = null
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@GroupId:** Group identifier.

**@TermSetId:** Term set identifier.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [GetChangeTimeResultSet](#)

### 3.1.5.26 proc\_ECM\_GetDescendentTerms

The **proc\_ECM\_GetDescendentTerms** stored procedure is called to retrieve all the **descendant** terms of a given term. If the term store partition does not exist or does not contain the specified term set, or the term set does not contain the specified term, no data is returned. Otherwise the six result sets listed in the results sets section for this procedure MUST be returned. If the term has no descendant terms, the six result sets MUST all be empty. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_GetDescendentTerms (
  @PartitionId uniqueidentifier
  ,@TermSetId int
  ,@TermId int
);

```

**@PartitionId:** The identifier of the term store partition that the term set containing the term belongs to. This parameter MUST NOT be NULL.

**@TermSetId:** The internal identifier (1) of the term set which the term belongs to. This parameter MUST NOT be NULL.

**@TermId:** The internal identifier of the term whose descendant terms are to be retrieved. This parameter MUST NOT be NULL.

**Return Values:** An integer which MUST be zero.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTerm.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

### 3.1.5.27 proc\_ECM\_GetGroupByGuid

The **proc\_ECM\_GetGroupByGuid** stored procedure is called to retrieve the properties of a group in a term store partition. The stored procedure MUST return the three result sets listed in the Result Sets section for this procedure. If either the term store partition or the group does not exist, the three result sets MUST be returned empty. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetGroupByGuid (  
  @GroupGuid uniqueidentifier  
  ,@PartitionId uniqueidentifier  
);
```

**@GroupGuid:** The identifier of the group to be retrieved. This parameter MUST NOT be NULL.

**@PartitionId:** The identifier of the term store partition that the group belongs to. This parameter MUST NOT be NULL.

**Return Values:** An integer which MUST be 0.

#### Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

### 3.1.5.28 proc\_ECM\_GetGroups

The **proc\_ECM\_GetGroups** stored procedure is called to retrieve the properties of all the specified groups in the term store partition. The stored procedure returns the three result sets listed in the Result Set section for this procedure. If the term store partition does not exist, or none of the specified groups exist, the three result sets MUST be returned empty. If some or all of the specified groups exist, the result sets MUST contain data for those groups. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetGroups (  
  @GroupIdList varchar(max)  
  ,@PartitionId uniqueidentifier  
);
```

**@GroupIdList:** A string containing a list of group internal identifiers (1). This parameter MUST NOT be NULL. This parameter is formatted as a sequence of tokens separated by a backslash (\) and MUST conform to the following ABNF definition:

```
@GroupIdList = iid *("\ " iid)...iid = 1*10DIGIT; internal identifier
```

For example, "1\2\3\4" is a valid string that represents the internal identifiers (1) 1, 2, 3, and 4.

**@PartitionId:** The identifier of the term store partition that the group belongs to. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

### 3.1.5.29 proc\_ECM\_GetPackage

The **proc\_ECM\_GetPackage** stored procedure is called to retrieve the package of a content type in the term store partition. If the term store partition does not exist or it does not contain the specified content type package, no data is returned. A match MUST produce a corresponding record in **proc\_ECM\_GetPackage.ResultSet0**. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetPackage (  
    @PartitionId uniqueidentifier  
    ,@Id varchar(512)  
    ,@Type uniqueidentifier  
);
```

**@PartitionId:** The identifier of the term store partition. The value MUST NOT be NULL.

**@Id:** The identifier of the content type. The value MUST NOT be NULL.

**@Type:** The value MUST be "B4AD3A44-D934-4C91-8D1F-463ACEADE443".

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [GetPackageResultSet](#)

### 3.1.5.30 proc\_ECM\_GetPackagesInformation

The **proc\_ECM\_GetPackagesInformation** stored procedure is called to get information about what content type packages have been modified in a term store partition since the last update time. If any global change affecting content type packages has occurred since the last update time, all packages affected by it MUST also be returned. If the term store partition does not exist or there is no match, no data is returned. Every match MUST produce a corresponding record in **proc\_ECM\_GetPackagesInformation.ResultSet0** (section [3.1.5.30](#)). This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetPackagesInformation (  
    @PartitionId uniqueidentifier  
    ,@LastUpdateTime datetime  
);
```

**@PartitionId:** The identifier of the term store partition. The value MUST NOT be NULL.

**@LastUpdateTime:** The UTC time of last update.

**Return Values:** An integer that MUST be 0.

## Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_GetPackagesInformation.ResultSet0](#)

### 3.1.5.31 proc\_ECM\_GetServiceSettings

This stored procedure will retrieve the service settings for the partition represented by the partition identifier passed in. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetServiceSettings (  
    @PartitionId uniqueidentifier  
);
```

**@PartitionId:** Partition identifier of the partition for which service settings will be returned.

**Return Values:** An integer that MUST be 0.

## Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_GetServiceSettings.ResultSet0](#)

### 3.1.5.32 proc\_ECM\_GetSessionData

The **proc\_ECM\_GetChangesForListSync.ResultSet0** result contains MUST return a row for every change matching the search condition or 0 rows if there is no match. The fields of the result set are populated using values from the **ECMUsedTerms** table (section [2.2.5.8](#)), the **ECMChangeLog** table (section [2.2.5.2](#)), and the **ECMServiceSettings** table (section [2.2.5.12](#)). The T-SQL syntax for the result set is as follows. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetSessionData (  
    @PartitionId uniqueidentifier  
);
```

**@PartitionId:** The unique identifier of a term set. The value MUST be NULL if the change is not scoped inside a term set, such as language change, permission change, partition data import, partition data deletion, group change, or used term change.

**Return Values:** An integer which MUST be 0.

## Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_GetSessionData.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetSessionData.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetSessionData.ResultSet2](#)

This stored procedure MUST return a single row containing the GUID identifying the system group in the result set [proc\\_ECM\\_GetSessionData.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetSessionData.ResultSet4](#)

This stored procedure MUST return a single row containing the GUID identifying the orphaned terms term set in the result set [proc\\_ECM\\_GetSessionData.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetSessionData.ResultSet6](#)

### 3.1.5.33 proc\_ECM\_GetSortedChildTermIds

The **proc\_ECM\_GetSortedChildTermIds** stored procedure retrieves the identifiers of all the child terms of the specified parent term. The identifiers are retrieved from the **ECMTerm** table (section [2.2.5.5](#)) and returned in the result set identified in the result sets section for this procedure. They are sorted in alphabetical order of the term label of each term as defined in the **ECMTermLabel** table (section [2.2.5.6](#)). The term label is determined as follows: if the term has a default term label in the specified preferred locale, that default term label is used. Otherwise the default term label in the default locale of the term store partition is used. If the specified term store partition does not exist, or it does not contain the specified term set, or the parent term does not exist in the term set, an empty result set is returned. This query does not return any terms from the **ECMMergedTerm** table (section [2.2.5.9](#)). This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetSortedChildTermIds (  
  @PartitionId uniqueidentifier  
  ,@TermSetId int  
  ,@ParentTermId int  
  ,@Locale int  
);
```

**@PartitionId:** The identifier of the term store partition that the term set belongs to. This parameter MUST NOT be NULL.

**@TermSetId:** The internal identifier (1) of the term set which the term belongs to. This parameter MUST NOT be NULL.

**@ParentTermId:** The internal identifier (1) of the parent term whose child terms are to be retrieved. This parameter MUST NOT be NULL.

**@Locale:** The internal identifier (1) of the preferred locale. This parameter MUST NOT be NULL.

**Return Values:** An integer which MUST be zero.

#### Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_GetSortedChildTermIds.ResultSet0](#)

### 3.1.5.34 proc\_ECM\_GetTermInTermSetByGuid

The **proc\_ECM\_GetTermInTermSetByGuid** stored procedure searches for a term by its identifier in the specified term store partition and returns its properties. If the specified term set does not contain the term, or if the term is not defined in the **ECMTerm** (section [2.2.5.5](#)) or the **ECMMergedTerm** table (section [2.2.5.9](#)), no result sets are returned. If the term is successfully found, the result sets identified in the Result Sets section for this procedure are returned. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetTermInTermSetByGuid (  
  @PartitionId uniqueidentifier  
  ,@TermGUID uniqueidentifier  
  ,@TermSetId int  
);
```

**@PartitionId:** The identifier of the term store partition that the term belongs to, as defined in either the **ECMTerm** or **ECMMergedTerm** tables. This parameter MUST NOT be NULL.

**@TermGUID:** The identifier of the term that is to be retrieved. This parameter MUST NOT be NULL. This parameter is first matched with the **UniqueId** column of the **ECMTerm** table. If no match is found, it is matched with the **TermUniqueId** column of the **ECMMergedTerm** table. If a match is still not found, no result sets are returned.

**@TermSetId:** The internal identifier (1) of the term set that the term is a member of, as defined in the **ECMTermSetMembership** table (section [2.2.5.14](#)). This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

#### **Result Sets:**

The following result sets MUST be returned if the term corresponding to **@TermGUID** exists in the term set and term store partition:

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

### **3.1.5.35 proc\_ECM\_GetTerms**

The **proc\_ECM\_GetTerms** stored procedure is called to retrieve a set of terms with matching identifiers. All term information is returned, including term labels, descriptions, custom properties and hierarchy information. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetTerms (
  @PartitionId uniqueidentifier
  ,@TermIdList varchar(max)
  ,@IsGuidList bit = 0
  ,@IsXML bit = 0
  ,@GetTermSet bit = 0
  ,@LCID int = null
  ,@DefaultLcid int = null
  ,@TermCount int = null
  ,@IsAllTermsInTermset bit = 0
  ,@TermSetIntId int = null
  ,@IncludeIdPath bit = 0
  ,@IsOptimized bit = 0
);
```

**@PartitionId:** A partition identifier (1). This parameter MUST NOT be NULL.

**@TermIdList:** A set of term identifiers indicating which terms should be retrieved. If **@IsGuidList** is 0, this MUST be a list of internal identifiers (1) of a term separated by a backslash (\). If

**@IsGuidList** is 1, this MUST be a list of identifiers of a term separated by a backslash (\). This parameter has no effect if **@IsAllTermsInTermset** is 1.

**@IsGuidList:** A value specifying whether the **@TermIdList** parameter contains unique identifiers or internal identifiers (1). The value MUST be 0 or 1. If the value is 0, the values specified in the **@TermIdList** parameter MUST be integers. If the value is 1, the values specified in the **@TermIdList** parameter MUST be GUIDs. This parameter has no effect if **@IsAllTermsInTermset** is 1.

**@IsXML:** Reserved. The value MUST be 0.

**@GetTermSet:** A value indicating whether term set information should be retrieved. The value MUST be 0 or 1. If the value is 0, term set information is not retrieved. If the value is 1, term set information is retrieved.

**@LCID:** The LCID that indicates the language to be used for generating the Path column for the retrieved terms in **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.LcidNotNull.ResultSet9** (section [2.2.4.41](#)) only if **@LCID** is NOT NULL. If **@LCID** is NULL then **proc\_ECM\_GetTerms.ResultSet3** (section [2.2.4.16](#)) is retrieved instead.

**@DefaultLcid:** The LCID that indicates the language to be used for generating the Path column for the retrieved terms in **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.LcidNotNull.ResultSet9**. If **@DefaultLcid** is NULL and **@Lcid** is NOT NULL, **@Lcid** is used instead.

**@TermCount:** A value specifying the number of terms expected to be returned. The value MUST NOT exceed the number of identifiers in the **@TermIdList** parameter value. This parameter is used to retrieve merged terms if the initial set of results is not complete. This parameter has no effect if **@IsGuidList** is 0.

**@IsAllTermsInTermset:** A value indicating whether all terms will be retrieved from a specific term set. The value MUST be 0 or 1. If the value is 0 then the identifiers in the **@TermIdList** parameter are used, and the **@TermSetIntId** parameter has no effect. If the value is 1, the identifiers in the **@TermIdList** have no effect, and all terms from the term set identified by the **@TermSetIntId** parameter are retrieved.

**@TermSetIntId:** A term set internal identifier (1) used to identify a term set from which all terms will be retrieved. This value MUST NOT be NULL if **@IsAllTermsInTermset** is 1. This parameter has no effect if **@IsAllTermsInTermset** is 0.

**@IncludeIdPath:** A value indicating whether full term identifier paths are returned in the result sets. The value MUST be 0 or 1. If this value is 0 then no full term identifier path information is returned. If this value is NOT 0 then full term identifier path information is returned.

**@IsOptimized:** A value indicating which subset of the data will be returned. The value MUST be 0 or 1. If the value is 1, the **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.IncludeTermSet.ResultSet5** (section [2.2.4.14](#)) result set containing child term information MUST NOT be returned, but the **proc\_ECM\_GetTermSetByGuid.ResultSet4** (section [2.2.4.20](#)) result set MUST be returned. If the value is 0, the **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.IncludeTermSet.ResultSet5** result set containing child term information MUST be returned, but the **proc\_ECM\_GetTermSetByGuid.ResultSet4** result set MUST NOT be returned. The exact result sets returned are listed in the result sets section.

**Return Values:** An integer which MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier (1).
- **@TermIdList:** A term identifier list.
- **@GetTermSet:** 1.
- **@IsOptimized:** 0.
- **@LCID:** A valid LCID.

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.LcidNotNull.ResultSet9](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier (1).
- **@TermIdList:** A term identifier list.
- **@GetTermSet:** 1.
- **@IsOptimized:** 0.

- **@LCID**: NULL.

This stored procedure MUST return a `proc_ECM_GetGroupByGuid.ResultSet0`

This stored procedure MUST return a `proc_ECM_GetGroupByGuid.ResultSet1`

This stored procedure MUST return a `proc_ECM_GetGroupByGuid.ResultSet2`

This stored procedure MUST return a `proc_ECM_GetTermSetByGuid.ResultSet3`

This stored procedure MUST return a `proc_ECM_GetTermSetByGuid.ResultSet4`

This stored procedure MUST return a `proc_ECM_GetTermSetByGuid.ResultSet5`

This stored procedure MUST return a `proc_ECM_GetTerms.ResultSet0`

This stored procedure MUST return a `proc_ECM_GetTerms.ResultSet1`

This stored procedure MUST return a `proc_ECM_GetTerms.ResultSet2`

This stored procedure MUST return a `proc_ECM_GetTerms.ResultSet3`

This stored procedure MUST return a `proc_ECM_GetTerms.ResultSet4`

This stored procedure MUST return a `proc_ECM_GetTerms.proc_ECM_GetTerms.IncludeTermSet.ResultSet5`

### 3.1.5.36 `proc_ECM_GetTermSets`

The **`proc_ECM_GetTermSets`** stored procedure searches for term sets in a term store partition and returns information about these term sets based on the specified flags. The list of term sets to search for is specified by the **`@TermSetIdList`** parameter. The format of this list is described by the **`@IsGuidList`** and **`@IsXML`** parameters. Every matching term set MUST produce a corresponding record in the **`proc_ECM_GetTermSetByGuid.ResultSet3`** result set (section [2.2.4.21](#)). This result set MUST always be returned. Additional result sets MUST be returned if either the **`@GetGroup`** or **`@GetRootTerms`** parameters are set to 1. Any term sets not found in the term store partition as defined in the **`ECMTermSet`** table (section [2.2.5.3](#)) MUST be ignored. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetTermSets (
  @PartitionId uniqueidentifier
  ,@TermSetIdList varchar(max)
  ,@IsGuidList bit = 0
  ,@IsXML bit = 0
  ,@GetGroup bit = 0
  ,@GetRootTerms bit = 1
  ,@SkipKeywordTermSet bit = 0
);
```

**@PartitionId**: The identifier of the term store partition that contains the term sets specified by **`@TermSetIdList`**. If this parameter is NULL, or the corresponding term store partition is not found, the result sets MUST be empty.

**@TermSetIdList**: A string that contains the list of term set identifier tokens for which information is to be retrieved. The format of this string is described by the **`@IsXML`** parameter, and the format of the identifier tokens is described by the **`@IsGuidList`** parameter. If the **`@TermSetIdList`**

parameter is NULL, or if none of the corresponding term sets are found, the result sets MUST be empty.

**@IsGuidList:** An optional bit flag that specifies the format of each identifier token in **@TermSetIdList**. This parameter is 0 by default. If this parameter is set to 1, the identifier tokens MUST be GUID identifiers and are matched with the term set identifiers. The **ABNF** definition is as follows:

```
idtoken = "{" guid "}" / guid
guid    = 8HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 4HEXDIG "-" 12HEXDIG
```

If this parameter is set to 0, the identifier tokens MUST be integer identifiers and are matched with the term set internal identifiers (1). The ABNF definition is:

```
idtoken = 1*10DIGIT
```

**@IsXML:** An optional bit flag that specifies the format of the string parameter **@TermSetIdList**. This parameter is 0 by default. If this parameter is set to 1, **@TermSetIdList** is formatted as an XML-like string of **row** elements with **id** attributes containing identifier tokens, and MUST conform to the following ABNF definition:

```
@TermSetIdList = *("<row id=" DQUOTE idtoken DQUOTE "/>")
```

The format for **idtoken** is specified in the **@IsGuidList** parameter description. For example, '<row id="1"/><row id="2"/><row id="3"/><row id="4"/>' represents the identifier tokens 1, 2, 3 and 4.

If this parameter is set to 0, **@TermSetIdList** is formatted as a sequence of identifier tokens separated by a backslash (\), and MUST conform to the following ABNF definition:

```
@TermSetIdList = idtoken *("\ " idtoken)
```

The format for **idtoken** is specified in the **@IsGuidList** parameter description. For example, "1\2\3\4" is a valid string that represents the identifier tokens 1, 2, 3, and 4.

**@GetGroup:** An optional bit flag that MUST be set to 1 to return additional result sets describing the groups that contain the term sets. When this parameter is 1, the additional result sets **proc\_ECM\_GetGroupByGuid.ResultSet0** (section [2.2.4.5](#)), **proc\_ECM\_GetGroupByGuid.ResultSet1** (section [2.2.4.4](#)), and **proc\_ECM\_GetGroupByGuid.ResultSet2** (section [2.2.4.3](#)) MUST be returned BEFORE any of the result sets identified in the Result Sets section for this procedure. These result sets contain all the distinct groups that all the term sets in **@TermSetIdList** belong to. If this parameter is 0, these result sets MUST NOT be returned. This parameter is 0 by default.

**@GetRootTerms:** An optional bit flag that MUST be set to 1 to return the root terms of all the term sets in **proc\_ECM\_GetTermSetByGuid.ResultSet3** (section [2.2.4.21](#)). In that case the additional result set **proc\_ECM\_GetTermSetByGuid.ResultSet4** (section [2.2.4.20](#)) MUST be returned containing one record for every matching root term. If this parameter is set to "0", this result set MUST NOT be returned. This parameter is "1" by default.

**@SkipKeywordTermSet:** Indicates whether keywords term set is skipped or not.

**Return Values:** An integer that MUST be "0".

**Result Sets:**

This stored procedure MUST return a `proc_ECM_GetTermSetByGuid.ResultSet3`

This stored procedure MUST return a `proc_ECM_GetTermSetByGuid.ResultSet4`

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

### 3.1.5.37 `proc_ECM_GetTermSetsByLabel`

The `proc_ECM_GetTermSetsByLabel` stored procedure searches for all term sets in a term store partition that contain terms with term labels matching those specified in the `@Labels` parameter (hereafter "specified labels"). The term labels are matched using the [\[MSDN-TSQL-Ref\]](#) LIKE condition. The following additional constraints apply:

- Every matched term set MUST contain at least one term with a term label that matches each specified label. If the term set has terms that match only some of the specified labels, that term set MUST NOT be matched.
- Every matched term set MUST be available for tagging (the `AvailableForTagging` field in the `ECMTermSet` table (section [2.2.5.3](#)) must be set to 1 for the term set).
- The matching terms in each matched term set MUST NOT be deprecated (the `IsDeprecated` field in the `ECMTerm` table (section [2.2.5.5](#)) must be set to 0 for the term).
- If the `@LCID` parameter is not NULL, only term labels in the corresponding language will be matched. If this parameter is NULL, all term labels in all languages will be matched.

If multiple term sets contain terms that match every specified label, all of those term sets MUST be returned. If the term store partition is not found, or if no term set contains terms that match all of the specified labels, the result sets MUST be empty. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetTermSetsByLabel (
  @PartitionId uniqueidentifier
  ,@Labels nvarchar(max)
  ,@LCID int = null
);
```

**@PartitionId:** The identifier of the term store partition from which the term sets MUST be returned. If the identifier is NULL, or if the term store partition is not found, the result sets will all be empty.

**@Labels:** A string containing tokens delimited by semicolons as defined by the following ABNF:

```
@Labels = token *(";" token)
```

Where *token* is a Unicode string that MUST NOT contain the following characters:

- Semicolon (;)
- Double quote (")
- Left angle bracket (<)
- Right angle bracket (>)

- Vertical bar (|)
- Ampersand (&)

Each token is interpreted as the starting portion of a term label on which to search for term sets. For example, to match the term label "abcd", valid tokens are "a", "ab", "abc", and "abcd". This parameter MUST NOT be NULL.

**@LCID:** The LCID that identifies the language in which to perform the matches. If this parameter is omitted or NULL, the search will be performed across all term labels in all languages. If this parameter is specified, the protocol client MUST provide a valid LCID from the WorkingLanguageId column of the **ECMLanguage** table (section [2.2.5.7](#)).

**Return Values:** An integer which MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

### 3.1.5.38 proc\_ECM\_ImportPartitionData

This stored procedure is called to bulk import data for a term store partition. Data is imported into the following tables for the term store partition:

- **ECMServiceSettings** (section [2.2.5.12](#))
- **ECMLanguage** (section [2.2.5.7](#))
- **ECMGroup** (section [2.2.5.11](#))
- **ECMTermSet** (section [2.2.5.3](#))
- **ECMTerm** (section [2.2.5.5](#))
- **ECMTermSetMembership** (section [2.2.5.14](#))
- **ECMTermLabel** (section [2.2.5.6](#))
- **ECMTermDescription** (section [2.2.5.13](#))
- **ECMTermProperty** (section [2.2.5.10](#))
- **ECMPermission** (section [2.2.5.15](#))
- **ECMMergedTerm** (section [2.2.5.9](#))
- **ECMUsedTerms** (section [2.2.5.8](#))
- **ECMPackage** (section [2.2.5.4](#))

After this stored procedure completes successfully, the log of changes MUST contain an entry with PartitionId as **@PartitionId**, **ObjectType** as 4, **ChangeType** as 8, **ChangeTime** as current UTC time, and the rest as NULL to note that data for the specified term store partition was imported. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_ImportPartitionData (  
    @PartitionId uniqueidentifier  
    ,@FolderPath nvarchar(1000)  
);
```

**@PartitionId:** Identifier of the term store partition for which data is being imported. This parameter MUST NOT be NULL.

**@FolderPath:** Local path on the data base server or **UNC** path to the folder that contains the files to be imported.

**Return Values:** An integer which MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.39 proc\_ECM\_LogChange

This stored procedure is called at the end of a transaction to create entries in the **ECMChangeLog** table (section [2.2.5.2](#)) for each logged add/update/delete operation performed in that transaction. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_LogChange (  
);
```

**Return Values:** An integer which MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.40 proc\_ECM\_MergeTerm

This stored procedure is called to merge a term into a **target term** thus creating a **merged term**.

If terms being merged have different parent term sets then the following apply:

- If a default label for a particular language for the target term is also the default label for the same language for a sibling of the original term, the error **ECMTaxonomyError\_19** described in section [2.2.7](#) MUST be raised.
- The original term MUST NOT be a reuse of the target term and vice versa.

If the terms being merged have the same parent term set then the following apply:

- The terms MUST be siblings.
- If a default label for a particular language for a child term of the target term is also the default label for the same language for a child term of the original term then the error **ECMTaxonomyError\_19** described in section [2.2.7](#) MUST be raised.

The terms being merged MUST be source terms and MUST NOT belong to the orphaned terms term set.

When the terms are merged successfully the following changes MUST occur:

- In term sets where the original term and target term are siblings the original term is deleted.
- In term sets which contain only the original term, the original term is not deleted and becomes a **reused term** of the target term.
- The target term is the new parent term of all child terms of the original term in all the parent term sets of original term.
- The term labels of the original term become the non-default labels of the target term.
- All usages of the original term in **Sites** as indicated in the **ECMUsedTerms** table (section [2.2.5.8](#)) are replaced by the target term.
- The merge is recorded by adding the GUID of the original term to the **MergedIdList** column of the **ECMTerm** table (section [2.2.5.5](#)) for the target term and by creating an entry in the **ECMMergedTerm** table (section [2.2.5.9](#)).

After this stored procedure is called and the transaction is committed there MUST be a new entry in the [ECMChangeLog](#) table with a **ChangeType** of **OperationTypeMerge** and an **ObjectType** of **ItemTypeTerm** for every parent term set of the original term. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure.

This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_MergeTerm (  
  @PartitionId uniqueidentifier  
  ,@TermId int  
  ,@TargetId int  
  ,@LastModifiedTime datetime = null  
  ,@TargetLastModifiedTime datetime = null  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the original term being merged. If a corresponding valid term does not exist then error ECMTaxonomyError\_10 described in section [2.2.7](#) MUST be raised. This parameter MUST NOT be NULL. Value MUST NOT be the same as the **@TargetId** parameter.

**@TargetId:** Internal identifier (1) of the target of the merge. If a corresponding valid term does not exist then error ECMTaxonomyError\_10 described in section [2.2.7](#) MUST be raised. This parameter MUST NOT be NULL.

**@LastModifiedTime:** The date and time in UTC format that indicates whether the protocol client has the latest view of the original term. If the value is not NULL and is not a future time compared to the LastModifiedTime column of the **ECMTerm** table for the original term then the error **ECMTaxonomyError\_11** described in section [2.2.7](#) MUST be raised.

**@TargetLastModifiedTime:** The date and time in UTC format that indicates whether the protocol client has the latest view of the target term. If the value is not NULL and is not a future time compared to the LastModifiedTime column of the **ECMTerm** table for the target term then the error **ECMTaxonomyError\_11** described in section [2.2.7](#) MUST be raised

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
-1	The merge operation failed because of errors <b>ECMTaxonomyError_10</b> , <b>ECMTaxonomyError_11</b> , or <b>ECMTaxonomyError_19</b> described in section <a href="#">2.2.7</a>
<b>ECMUsedTerms.@@error</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.41 proc\_ECM\_MoveTermSetMembership

This stored procedure moves the specified term from the source term set to become a member of the specified target term set within the specified term store partition. It also moves all descendant terms of the specified term in the source term set to become members of the specified target term set accordingly. It MUST NOT be used to move a term within a term set and MUST take NO action when source term set and specified target term set are the same. If moving the specified term as a member of the specified target term set would result in multiple terms with same default label and parent term then the error **ECMTaxonomyError\_19** described in error section [2.2.7](#) MUST be raised. On success, and after **proc\_ECM\_LogChange** is called and the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeMove** and an **ObjectType** of **ItemTypeTerm**. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_MoveTermSetMembership (  
    @PartitionId uniqueidentifier  
    ,@TermSetId int  
    ,@TermId int  
    ,@ParentTermId int = 0  
    ,@AvailableForTagging bit = 0  
    ,@CustomSortOrder nvarchar(max) = null  
    ,@IsSource bit  
    ,@SourceTermSetId int  
    ,@SourceParentTermId int = 0  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermSetId:** Internal identifier (1) of the target term set to which a term is being moved to as a member. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term being moved. This parameter MUST NOT be NULL.

**@ParentTermId:** Internal identifier (1) of the target parent term for the term being moved. This parameter MUST NOT be NULL.

**@AvailableForTagging:** Indicates whether term being moved is available to be used by end users for tagging. This parameter MUST NOT be NULL.

**@CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the **UniqueId** column of the **ECMTerm** table (section [2.2.5.5](#)).

**@IsSource:** Indicates whether the term being added to the term set is a source term. This parameter MUST NOT be NULL.

**@SourceTermSetId:** Internal identifier (1) of the term set that the specified term is being moved from. This parameter MUST NOT be NULL.

**@SourceParentTermId:** Internal identifier (1) of the parent term of the term being moved. This parameter MUST NOT be NULL.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
<b>ECMTermSetMembership.@@error</b>	Type the return value description.
<b>-1</b>	The move operation failed because of error <b>ECMTaxonomyError_19</b> described in section <a href="#">2.2.7</a> .

#### Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_MoveTermSetMembership.ResultSet0](#)

### 3.1.5.42 proc\_ECM\_PopulateLabelsForDefaultLanguage

This stored procedure populates names of existing term sets and the corresponding default labels for the specified new default language within the specified term store partition. For term sets that do not have a name for the term set in the specified new language, it MUST append the name of the term set specified by the previous language. Thereafter it MUST verify that there are no term sets that share the same names in the same language. If such duplicates exist, then the error **ECMTaxonomyError\_22** described in error section [2.2.7](#) MUST be raised. Then for terms that do not have default labels for the specified new language it MUST copy the default label from the specified previous language. If, as a result of this operation, terms having the same default label and parent terms exist, then the error **ECMTaxonomyError\_23** described in error section [2.2.7](#) MUST be raised. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_PopulateLabelsForDefaultLanguage (  
    @PartitionId uniqueidentifier  
    ,@NewDefaultLanguageId int  
    ,@PreviousDefaultLanguage int  
    )  
    ;
```

**@PartitionId:** The term store partition identifier. This parameter MUST NOT be NULL.

**@NewDefaultLanguageId:** The LCID of the new default language. This parameter MUST NOT be NULL.

**@PreviousDefaultLanguage:** The LCID that identifies the previous default language. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST one of the following.

Value	Description
<b>0</b>	Default return value

Value	Description
-1	Population failed because either the error <b>ECMTaxonomyError_22</b> or <b>ECMTaxonomyError_23</b> described in <a href="#">2.2.7</a> was raised.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.43 **proc\_ECM\_RemoveApplicationLog**

This stored procedure is called to remove entries from the **ECMApplicationLog** (section [2.2.5.1](#)) used by content type syndication. It removes entries in the specified term store partition that are older than or equal to the specified time from the table described in section [2.2.5.1](#). This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_RemoveApplicationLog (
  @PartitionId uniqueidentifier
  ,@EndTime datetime
);
```

**@PartitionId:** Term store partition identifier. The parameter MUST NOT be NULL.

**@EndTime:** The UTC time. Log entries in the term store partition that are older than or equal to that time will be removed. The parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.44 **proc\_ECM\_RemoveUsedTermSets**

The **proc\_ECM\_RemoveUsedTermSets** stored procedure is called to remove all terms belonging to a term set in a term store partition from the **ECMUsedTerms** table (section [2.2.5.8](#)). This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_RemoveUsedTermSets (
  @PartitionId uniqueidentifier
  ,@TermSetId uniqueidentifier
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermSetId:** Term set identifier for the term set to which the terms being removed belong to. All terms in this term set MUST be removed from the **ECMUsedTerms** table. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.45 **proc\_ECM\_RetrieveApplicationLog**

This stored procedure is used to retrieve entries from the **ECMApplicationLog** (section [2.2.5.1](#)) used by content type syndication. It returns the oldest 1000 entries for the specified term store

partition from the table described in section [2.2.5.1](#), or all entries if there are less than 1000 in the specified term store partition. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_RetrieveApplicationLog (
  @PartitionId uniqueidentifier
);
```

**@PartitionId:** Term store partition identifier. The value MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_RetrieveApplicationLog.ResultSet0](#)

### 3.1.5.46 proc\_ECM\_RetrieveTableNames

This stored procedure is used to retrieve the names of all the user-defined tables in the database. The names of all the tables listed in [2.2.5](#) MUST be returned in the result set. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_RetrieveTableNames (
);
```

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_RetrieveTableNames.ResultSet0](#)

### 3.1.5.47 proc\_ECM\_SearchByLabel

The **proc\_ECM\_SearchByLabel** stored procedure is called to retrieve a set of terms that have matching term labels. All term information is returned, including all term labels, descriptions, and custom properties. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SearchByLabel (
  @PartitionId uniqueidentifier
  ,@Label nvarchar(255)
  ,@ResultSize int
  ,@TermSetId int = null
  ,@AnchorId int = null
  ,@LCID int = null
  ,@DefaultLabelOnly int = 0
  ,@IncludeIdPath int = 0
  ,@MatchOption smallint = 1
  ,@GetFullPath bit = 0
  ,@DefaultLcid int = null
  ,@IsOptimized bit = 0
  ,@GetAllLocalTermSets bit = 0
  ,@SiteGroupId int = null
  ,@CollationForLcid nvarchar(128) = null OUTPUT
  ,@CollationForDefaultLcid nvarchar(128) = null OUTPUT
);
```

**@PartitionId:** A partition identifier. This parameter MUST NOT be NULL.

**@Label:** A value specifying the term label value with which to search. The string matching is case insensitive.

**@ResultSize:** A value specifying the maximum number of terms to be retrieved. This value MUST be 0 or greater.

**@TermSetId:** A term set internal identifier (1) used to restrict the scope of the search. Only terms that are members of this specified term set will be retrieved. If this value is NULL then the terms will be retrieved from any term set.

**@AnchorId:** A term internal identifier (1) used to restrict the scope of the search. Only child terms of this specified term will be retrieved. If the term identified by the **@AnchorId** parameter value is not a member of the term set identified by the **@TermSetId** parameter then all result sets will be empty. If the **@TermSetId** parameter value is NULL then this parameter has no effect.

**@LCID:** The LCID that identifies the language in which to perform the matches. If this parameter is omitted or NULL, the search will be performed across all term labels in all languages. If this parameter is specified, only term labels in the corresponding language will be matched. If specified, the protocol client MUST provide a valid LCID from the **WorkingLanguageId** column of the **ECMLanguage** table (section [2.2.5.7](#)).

**@DefaultLabelOnly:** A value indicating whether the term search should be restricted to matching on default labels. If the value is 1 then only default labels will be matched against the **@Label** value. If the value is NOT 1 then any term labels will be matched against the **@Label** value.

**@IncludeIdPath:** A value indicating whether term GUID paths are returned in the result sets. If this value is 0, no term GUID path information is returned. If this value is NOT 0 and the **@GetFullPath** parameter value is NOT 0, term GUID path information is returned.

**@MatchOption:** A value indicating the type of string match to perform on the term label value. If this value is 1, the term label value MUST exactly match the value specified in the **@Label** parameter. If this value is NOT 1, the term label value MUST start with the value specified in the **@Label** parameter.

**@GetFullPath:** A value indicating whether **term label paths** are returned in the result sets. If this value is 0, no term label path information is returned. If this value is NOT 0 and the **@LCID** parameter value is NOT NULL, term label path information is returned.

**@DefaultLcid:** The LCID that indicates the language to be used for generating the Path column for the retrieved terms in **proc\_ECM\_GetTerms.proc\_ECM\_GetTerms.LcidNotNull.ResultSet9** (section [2.2.4.41](#)). The path contains term label value for the language specified by **@LCID**. When a term does not have a default term label for **@LCID**, the default term label for **@DefaultLcid** is used. This parameter has no effect if **@GetFullPath** is 0.

**@IsOptimized:** A value indicating whether a subset of the data will be returned. The value must be 0 or 1. If the value is 1, a subset of the result sets will be returned, as defined in the result sets section.

**@GetAllLocalTermSets:** A value indicating whether to include all local term sets in the search scope. If the **@TermSetId** is NOT NULL, this value is ignored. Otherwise, if this value is NOT 0, the search scope will include all the local term sets. If this value is 0, the search scope will only include the local term sets in the site collection term set group specified by the **@SiteGroupId** parameter value.

**@SiteGroupId:** The term set group internal identifier (1). If the **@GetAllLocalTermSets** parameter value is NOT 0, this value is ignored. Otherwise, if this value is NOT NULL, the search scope includes the local term sets under the site collection term set group specified by this value. If this value is NULL, the search scope does not include any local term sets.

**@CollationForLcid:** The collation for term label when the parameter **@Lcid** is NOT NULL.

**@CollationForDefaultLcid:** The collation for term label when the parameter **@Lcid** is NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

For the following combination of parameters:

- **@PartitionId:** 0c37852b-34d0-418e-91c6-2ac25af4be5b
- **@Label:** 1
- **@ResultSize:** 10
- **@LCID:** 1033
- **@IncludeIdPath:** 1
- **@MatchOption:** 0
- **@IsOptimized:** 1

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_SearchByLabel.Optimized.ResultSet5](#)

### 3.1.5.48 **proc\_ECM\_SearchByProperty**

The **proc\_ECM\_SearchByProperty** stored procedure is called to retrieve a set of terms that have a matching custom property. All term information is returned, including labels, descriptions, and all custom properties. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SearchByProperty (
  @PartitionId uniqueidentifier
  ,@ResultSize int
  ,@PropertyName nvarchar(255)
  ,@PropertyValue nvarchar(max) = null
  ,@TermSetId int = null
  ,@MatchOption smallint = 1
);
```

**@PartitionId:** A partition identifier. This parameter MUST NOT be NULL.

**@ResultSize:** A value specifying the maximum number of terms to be retrieved. This value MUST be 0 or greater.

**@PropertyName:** A value specifying the custom property name with which to search. If this value is NULL then all result sets will have 0 rows. The string matching MUST be case-insensitive.

**@PropertyValue:** A value specifying the custom property value with which to search. The string matching MUST be case-insensitive. If this value is NULL then the custom property value MUST be ignored and the match MUST only be performed on the custom property name provided in the **@PropertyName** parameter.

**@TermSetId:** A term set internal identifier (1) used to restrict the scope of the search. Only terms that are members of this specified term set MUST be retrieved. If this value is NULL then the terms MUST be retrieved from all term sets.

**@MatchOption:** A value indicating the type of string match to perform on the custom property value. If this value is 1, the custom property value MUST exactly match the value specified in the **@PropertyValue** parameter. If this value is NOT 1, the custom property value MUST start with the value specified in the **@PropertyValue** parameter. This parameter has no effect if **@PropertyValue** is NULL.

**Return Values:** An integer that MUST be zero.

#### **Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.IncludeTermSet.ResultSet5](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTerms.ResultSet4](#)

### 3.1.5.49 proc\_ECM\_SearchTermSetName

The **proc\_ECM\_SearchTermSetName** stored procedure is called to retrieve term sets that have a matching name. All term set information is returned, including the group and hierarchy information. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SearchTermSetName (  
    @PartitionId uniqueidentifier  
    ,@TermSetName nvarchar(255)  
    ,@LCID int  
);
```

**@PartitionId:** A partition identifier. This parameter MUST NOT be NULL.

**@TermSetName:** A value specifying the term set name with which to search. The string matching MUST be case-insensitive.

**@LCID:** An LCID that restricts the language in which the specified term set name will be matched. Protocol clients MUST provide a valid LCID from the **WorkingLanguageId** column of the **ECMLanguage** table (section [2.2.5.7](#)). This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

#### Result Sets:

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

### 3.1.5.50 proc\_ECM\_SelectPartitionData

This stored procedure is called to get all data from the table specified by the **@tableName** parameter in the term store partition specified by the **@PartitionId** parameter. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SelectPartitionData (  
    @PartitionId uniqueidentifier  
    ,@tableName nvarchar(100)  
    );
```

**@PartitionId:** Identifier for the term store partition from which data is to be selected. This parameter MUST NOT be NULL.

**@tableName:** Name of the table from which data is to be selected. Value MUST be one of the following:

- **ECMServiceSettings**
- **ECMLanguage**
- **ECMGroup**
- **ECMTermSet**
- **ECMTerm**
- **ECMTermSetMembership**
- **ECMTermLabel**
- **ECMTermDescription**
- **ECMTermProperty**
- **ECMPermission**
- **ECMMergedTerm**
- **ECMUsedTerms**
- **ECMPackage**

**Return Values:** An integer that MUST be 0.

**Result Sets:**

For the following combination of parameters:

- **@PartitionId:** Select **PartitionId** from **ECMTerm**.
- **@tableName: ECMTerm.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTerm.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMTermLabel.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermLabel.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMTermSet.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermSet.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMPackage.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPackage.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMLanguage.**

This stored procedure MUST return a [proc\\_ECM\\_GetSessionData.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMUsedTerms.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMUsedTerms.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMMergedTerm.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMMergedTerm.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMTermProperty.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermProperty.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMGroup.**

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMServiceSettings.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMServiceSettings.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMTermDescription.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermDescription.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMTermSetMembership.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermSetMembership.ResultSet0](#)

For the following combination of parameters:

- **@PartitionId:** A valid partition identifier.
- **@tableName: ECMPermission.**

This stored procedure MUST return a [proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPermission.ResultSet0](#)

### 3.1.5.51 proc\_ECM\_SetDefaultLanguage

The **proc\_ECM\_SetDefaultLanguage** stored procedure is called to set the default language for a term store. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SetDefaultLanguage (  
  @PartitionId uniqueidentifier  
  ,@WorkingLanguageId int  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@WorkingLanguageId:** The LCID that identifies the language specified to be set as default for the term store. This parameter MUST NOT be NULL. If the specified value is already the default language for the term store, **proc\_ECM\_SetDefaultLanguage** MUST NOT perform any further actions. Otherwise, **proc\_ECM\_SetDefaultLanguage** MUST mark the previous default language as non-default; on success, and after **proc\_ECM\_LogChange** is called and the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a

**ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeTermStore**. If the specified value is not a supported language for the term store, this stored procedure MUST return a value of -1 and raise error with identifier [2.2.7](#). If this stored procedure failed to populate names of existing term sets and the corresponding default labels for the language specified as the new default, this stored procedure MUST return a value of -1.

**Return Values:** An integer that MUST be in the following table.

Value	Description
-1	Error conditioned occurred. See error raised.
@@error	An unexpected error occurred.
0	Default return value.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.52 proc\_ECM\_SetPackage

The **proc\_ECM\_SetPackage** stored procedure is called to add or update a content type package. If **@IsPublished** is 0 and the package does not exist in the table, no data will be added or updated. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SetPackage (
  @PartitionId uniqueidentifier
  ,@Id varchar(512)
  ,@Type uniqueidentifier
  ,@IsPublished bit
  ,@ContentSize int
  ,@Content varbinary(max)
);
```

**@PartitionId:** The identifier of term store partition. The value MUST NOT be NULL.

**@Id:** The identifier of the content type.

**@Type:** The value MUST be "B4AD3A44-D934-4C91-8D1F-463ACEADE443".

**@IsPublished:** Specifies whether the content type is published.

**@ContentSize:** The size of the content type package content.

**@Content:** The content of the content type package.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.53 proc\_ECM\_SetServiceSettings

This stored procedure will either update or insert service settings for a given partition identifier. If service settings exist for the partition id, they MUST be updated, and if service settings do not exist, they MUST be created. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_SetServiceSettings (
```

```

@PartitionId uniqueidentifier
,@Settings nvarchar(max)
);

```

**@PartitionId:** The partition identifier of the partition that will have its service settings updated or inserted into.

**@Settings:** The setting XML that MUST be formatted to match the XSD [\[MS-EMMWCF\]](#) **ServiceSettingsSettingsXmlDoc**.

**Return Values:** 0 for success, or an integer that MUST be in the following table.

Value	Description
@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.54 proc\_ECM\_UpdateGroup

This stored procedure is used to modify the properties of a term set group in the term store. If this stored procedure runs successfully (return code 0) the term set group identified by **@Id** MUST be updated with **@Name**, **@Description**, and the current time as **LastModifiedTime**. Also, on success, after [proc\\_ECM\\_LogChange](#) has been called and the transaction has been committed, an entry in ECMChangeLog MUST have been added with the **@PartitionId** as **PartitionId**, the term group identifier that corresponds to the internal identifier (1) **@Id** as **GroupUniqueId** and **ObjectUniqueId**, internal identifier (1) as **ObjectId**, 3 (**ItemTypeGroup**) as **ChangeType**, 2 (**OperationTypeEdit**) as **ChangeType**, current time as **ChangeTime** and NULL as **TermSetId**, **ChangeData**, and **ModifiedBy**. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdateGroup (
@PartitionId uniqueidentifier
,@Id int
,@Name nvarchar(255)
,@Description nvarchar(1000) = null
,@LastModifiedTime datetime = null
);

```

**@PartitionId:** Partition identifier of term group being updated.

**@Id:** Internal identifier (1) of term group being updated.

**@Name:** Term group name that MUST be stored when this stored procedure completes.

**@Description:** Term group description that MUST be stored when this stored procedure completes.

**@LastModifiedTime:** The last modified time of term group that is being updated in UTC format or NULL.

**Return Values:** An integer that MUST be in the following table.

Value	Description
-1	MUST be returned if <b>Id</b> does not exist or if the <b>Id</b> exists and but the <b>LastModifiedTime</b> of the existing row is NULL. MUST be returned if <b>@LastModifiedTime</b> is an older date than the current row <b>LastModifiedTime</b> .
<b>@@error</b>	Type the return value description.
0	Default return value.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.55 proc\_ECM\_UpdatePermission

This stored procedure is called to update the ACL associated with a user, specified by the **@PrincipalName** parameter, for the term store or term set group. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdatePermission (
  @PartitionId uniqueidentifier
  ,@GroupId int
  ,@PrincipalName nvarchar(255)
  ,@DisplayName nvarchar(255)
  ,@Rights int
  ,@AddToLog bit = 0
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@GroupId:** Internal identifier (1) of the term group for which ACL is being updated. If nonzero and a corresponding term group does not exist then the error **ECMTaxonomyError\_15** described in error section [2.2.7](#) MUST be raised.

**@PrincipalName:** The name of the user whose permissions or full name (or both) is being updated. If the **@GroupId** parameter is 0 then permissions to the term store is updated for this user.

**@DisplayName:** The display name of the user whose permissions to the term group or term store is being updated.

**@Rights:** Rights. Existing or updated rights of the user to the term store or a term group. Value MUST NOT be NULL. Value MUST be one of the values described in the column with name Rights in **ECMPermission** table (section [2.2.5.15](#)).

**@AddToLog:** Indicates whether this action is noted in the log of changes. If the value is 1 and the value of the **@GroupId** parameter is 0 then after this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeTermStore**. If the value is 1 and the value of the **@GroupId** parameter is not 0 then after this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** table with a **ChangeType** of **OperationTypeEdit** and an **ObjectType** of **ItemTypeGroup**.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
-1	The update failed because the error <b>ECMTaxonomyError_15</b> described in error section <a href="#">2.2.7</a> was raised.
@@error	Type the return value description.
ECMGroup.@@error	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.56 proc\_ECM\_UpdateTerm

This stored procedure is called to update information for a term in the table described in section [2.2.5.5](#). The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdateTerm (
  @PartitionId uniqueidentifier
  ,@Id int
  ,@Owner nvarchar(255)
  ,@UseCount int = 0
  ,@IsDeprecated bit = 0
  ,@LastModifiedTime datetime = null
  ,@AnyDefaultLabelChanged bit = 0
  ,@AddToLog bit = 1
  ,@ChangeData nvarchar(max) = null
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@Id:** Internal identifier (1) of the term being updated. If a corresponding valid term does not exist then error **ECMTaxonomyError\_10** described in section [2.2.7](#) MUST be raised. This parameter MUST NOT be NULL.

**@Owner:** Existing or updated name for the owner of the term. Value MUST NOT be NULL.

**@UseCount:** Reserved. Value of this parameter MUST be 0.

**@IsDeprecated:** Existing or updated value that indicates whether the term is a deprecated term.

**@LastModifiedTime:** The date and time in Coordinated Universal Time (UTC) format that indicates whether the protocol client has the latest view of the term being updated. If the value is not NULL and is not a future time compared to the **LastModifiedTime** column of the **ECMTerm** table (section [2.2.5.5](#)) for the term then the error **ECMTaxonomyError\_11** described in section [2.2.7](#) MUST be raised.

**@AnyDefaultLabelChanged:** Indicates whether a default label was modified for the term being updated. If the value of this parameter is 1 and the **@AddToLog** parameter is 1 then the log of changes MUST contain an entry with **PartitionId** as **@PartitionId**, **GroupUniqueId** as the identifier of the term's group, **TermSetUniqueId** as the identifier of the term's term set, **ObjectUniqueId** as the identifier of the term, **ObjectId** as the internal identifier of the term, **ObjectType** as **ItemTypeTerm**, **ChangeType** as **OperationTypePathChange**, **ChangeTime** as current UTC time, and the rest as NULL, noting that there was a change in the default label based path of term.

**@AddToLog:** Indicates whether this action is noted in the log of changes. If the value of this parameter is 1 then after this stored procedure is called, the log of changes MUST contain an entry with **PartitionId** as **@PartitionId**, **GroupUniqueId** as the identifier of the term's group, **TermSetUniqueId** as the identifier of the term's term set, **ObjectUniqueId** as the identifier of the term, **ObjectId** as the internal identifier of the term, **ObjectType** as **ItemTypeTerm**, **ChangeType** as **OperationTypeEdit**, **ChangeTime** as current UTC time, and the rest as NULL to note that the term specified by the **@Id** parameter was updated.

**@ChangeData:** Details of the term update. <12>

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
-1	The update failed because of errors <b>ECMTaxonomyError_10</b> or <b>ECMTaxonomyError_11</b> described in section <a href="#">2.2.7</a>
<b>@@error</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.57 proc\_ECM\_UpdateTermDescription

This stored procedure is called to update the description of a term. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdateTermDescription (
  @PartitionId uniqueidentifier
  ,@TermId int
  ,@LCID int
  ,@Description nvarchar(1000)
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term whose description is being updated. The parent term set of the term MUST NOT be the orphaned terms term set. This parameter MUST NOT be NULL.

**@LCID:** The LCID that indicates the language of the description. Protocol client MUST provide a valid LCID from the WorkingLanguageId column of the **ECMLanguage** table (section [2.2.5.7](#)). This parameter MUST NOT be NULL.

**@Description:** New description of the term. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.58 proc\_ECM\_UpdateTermProperty

This stored procedure is called to update a custom term property in the table described in section [2.2.5.10](#). The initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdateTermProperty (
  @PartitionId uniqueidentifier
  ,@TermId int
  ,@TermSetId int
  ,@PropertyName nvarchar(255)
  ,@PropertyValue nvarchar(max)
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term whose custom property is being updated. The parent term set of the term MUST NOT be the orphaned terms term set.

**@TermSetId:** Internal identifier (1) of the term set to which the term belongs. <13>

**@PropertyName:** Name of the custom term property being updated.

**@PropertyValue:** New value for the custom term property specified by the @PropertyName parameter. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.59 proc\_ECM\_UpdateTermSet

This stored procedure is called to update properties of a term set in the **ECMTermSet** table (section [2.2.5.3](#)). This term set MUST not use the same name in any working language as any other existing term set in the same group; otherwise error **ECMTaxonomyError\_21** described in error section [2.2.7](#) MUST be thrown. The Initialization routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. After this stored procedure is called and the transaction is committed there MUST be a new entry in the **ECMChangeLog** (section [2.2.5.2](#)) table with a **ChangeType** of either **OperationTypeMove** or **OperationTypeEdit** and an **ObjectType** of **ItemTypeTermSet**. The **ChangeType** of **OperationTypeMove** MUST be logged if the value of the **@GroupId** parameter is different from the internal identifier (1) of the parent term set group of the term set being updated. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdateTermSet (
  @PartitionId uniqueidentifier
  ,@Id int
  ,@Name nvarchar(max)
  ,@Description nvarchar(1000) = null
  ,@Owner nvarchar(255)
  ,@CustomSortOrder nvarchar(max) = null
  ,@IsOpen bit = 0
  ,@AvailableForTagging bit = 0
  ,@Stakeholders nvarchar(1000) = null
  ,@Contact nvarchar(320) = null
  ,@LastModifiedTime datetime = null
  ,@GroupId int
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@Id:** Internal identifier (1) of the term set being updated. If a corresponding valid term set does not exist then error **ECMTaxonomyError\_8** described in section [2.2.7](#) MUST be raised. This parameter MUST NOT be NULL.

**@Name:** Existing or updated names of the term set being updated. The value is a sequence of LCID|name separated by semicolon. The name value in the sequence MUST NOT exceed 255 characters, and it MUST NOT contain the following characters:

- Semicolon (;)
- Double quotes (")
- Left angle bracket (<)
- Right angle bracket (>)
- Vertical bar (|)
- Ampersand (&)
- \t

**@Description:** Existing or updated description of the term set.

**@Owner:** Existing or updated name of the user who owns the term set. This parameter MUST NOT be NULL.

**@CustomSortOrder:** Existing or updated sort order to be applied to the child terms of this term set. If not NULL, the sort order MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid identifier from the **UniqueId** column of the **ECMTerm** table (section [2.2.5.5](#)).

**@IsOpen:** Existing or updated value that indicates whether the term set is an open term set.

**@AvailableForTagging:** Existing or updated value that indicates whether this term set is available for use by end users for tagging

**@Stakeholders:** Existing or updated names of users in the organization that should be notified before any major changes are made to the term set. If not NULL, the names provided MUST be separated by a semicolon.

**@Contact:** Existing or updated e-mail address of the user to who term suggestion and feedback is to be provided.

**@LastModifiedTime:** The date and time in Coordinated Universal Time (UTC) format that indicates whether the protocol client has the latest view of the term set being updated. If the value is not NULL and is not a future time compared to the **LastModifiedTime** column of the **ECMTermSet** table (section [2.2.5.3](#)) for the term set then the error **ECMTaxonomyError\_9** described in section [2.2.7](#) MUST be raised

**@GroupId:** Existing or updated internal identifier (1) of the parent group of the term set. If this value is updated then the log of changes MUST contain an entry with **PartitionId** as **@PartitionId**, **GroupUniqueId** as the identifier of the term set's group, **TermSetUniqueId** as the identifier of the term set, **ObjectUniqueId** as the identifier of the term set, **ObjectId** as the internal identifier of the term set, **ObjectType** as **ItemTypeTermSet**, **ChangeType** as **OperationTypeMove**, **ChangeTime** as current UTC time, and the rest as NULL, noting that the term set was moved to a different term group. Otherwise the log of changes MUST contain an entry with **PartitionId** as **@PartitionId**, **GroupUniqueId** as the identifier of the term set's group, **TermSetUniqueId** as the identifier of the term set, **ObjectUniqueId** as the identifier of the term set, **ObjectId** as the

internal identifier of the term set, **ObjectType** as **ItemTypeTermSet**, **ChangeType** as **OperationTypeEdit**, **ChangeTime** as current UTC time, **ChangeData** as specified in [MS-EMMWCF] section 2.2.4.11, and the rest as NULL, to note that an edit operation was performed on the term set.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
-1	The update failed because of errors <b>ECMTaxonomyError_8</b> or <b>ECMTaxonomyError_9</b> described in section <a href="#">2.2.7</a>
<b>ECMTermSet.@@error</b>	Type the return value description.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.60 **proc\_ECM\_UpdateTermSetMembership**

This stored procedure updates term set membership properties for the specified term belonging to the specified term set within the specified term store partition. It also handles moving a term within the specified term set if a parent term's internal identifier (1) is different from the previous one is specified. If moving the specified term within the term set would result in multiple terms with same default label and parent term then the error **ECMTaxonomyError\_19** described in error section [2.2.7](#) MUST be raised. The **Initialization** routine specified in step 2 of section [3.1.5](#) MUST be executed before calling this stored procedure. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UpdateTermSetMembership (
  @PartitionId uniqueidentifier
  ,@TermSetId int
  ,@TermId int
  ,@ParentTermId int = 0
  ,@AvailableForTagging bit = 0
  ,@CustomSortOrder nvarchar(max) = null
  ,@IsSource bit
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@TermSetId:** Internal identifier (1) of the term set to which the specified term belongs as a member. This parameter MUST NOT be NULL.

**@TermId:** Internal identifier (1) of the term being updated. This parameter MUST NOT be NULL.

**@ParentTermId:** Internal identifier (1) of the parent term of the term being updated. This parameter MUST NOT be NULL. If the specified parent term differs from the previous parent term of the specified term, the stored procedure MUST move the specified term to become a child term of the specified parent term and MUST also move all the descendant terms of the specified term accordingly. In this case, on success, and after **proc\_ECM\_LogChange** is called and the transaction is committed, there MUST be a new entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a **ChangeType** of **OperationTypeMove** and an **ObjectType** of **ItemTypeTerm**.

**@AvailableForTagging:** Indicates whether term being updated is available to be used by end users for tagging. This parameter MUST NOT be NULL.

**@CustomSortOrder:** Sort order to be applied to the child terms of the term being added. By default the child terms will be sorted alphabetically by their default label. If not NULL, the sort order

MUST be of the form identifier<sub>1</sub>:identifier<sub>2</sub>... where identifier<sub>n</sub> is a valid GUID from the UniqueId column of the **ECMTerm** table (section [2.2.5.5](#)).

**@IsSource:** Indicates whether the term being updated is a source term. This parameter MUST NOT be NULL. If set to 1 and the previous value for the specified term was zero, the stored procedure MUST set any instance of the specified term belonging to term set other than the specified one as NOT a source term.

**Return Values:** 0 for success or an integer that MUST be in the following table.

Value	Description
<b>ECMTermSetMembership.@@error</b>	Type the return value description.
-1	The update operation failed because of error <b>ECMTaxonomyError_19</b> described in section <a href="#">2.2.7</a> .

**Result Sets:**

This stored procedure MUST return [proc\\_ECM\\_MoveTermSetMembership.ResultSet0](#)

### 3.1.5.61 proc\_ECM\_ClearUsedTerms

The **proc\_ECM\_ClearUsedTerms** stored procedure is called to clear the list of terms used in a site collection from a term store partition. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_ClearUsedTerms (  
  @PartitionId uniqueidentifier  
  ,@WebAppId uniqueidentifier  
  ,@ContentDatabaseId uniqueidentifier  
  ,@SiteGuid uniqueidentifier  
);
```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**@WebAppId:** The identifier of the Web application that contains the site collection in which the identified term is being used. This parameter MUST NOT be NULL.

**@ContentDatabaseId:** The identifier of the content database for the site collection in which the identified term is being used. This parameter MUST NOT be NULL.

**@SiteGuid:** The identifier for site collection for which the list of terms used MUST be cleared. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0 or in the following table.

Value	Description
<b>@@error</b>	An unexpected error occurred.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.62 proc\_ECM\_GetChangesForListSync

The **proc\_ECM\_GetChangesForListSync** stored procedure is called to retrieve details of term and term store changes used in a Web application occurring after a specified time. The details of changes MUST be associated with any of following operations: movement of a term, change of a term's path, merger of terms, and edits of a term where **ECMChangeLog:ChangedData** value is NOT NULL. Details of changes MUST be ordered in ascending order by term store partition identifier, then in descending numeric order by type of object for which the change has occurred, then in ascending order by site collection identifier and, finally in ascending order by term set identifier. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetChangesForListSync (  
    @WebAppId uniqueidentifier  
    ,@SinceTime datetime = null  
);
```

**@WebAppId:** Web application identifier for which to get changes. This parameter MUST NOT be NULL.

**@SinceTime:** The date and time value, in UTC, after which the changes that occurred MUST be retrieved. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetChangesForListSync.ResultSet0](#)

### 3.1.5.63 proc\_ECM\_ImportPackage

This stored procedure is used to store a content type package to the **ECMPackage** table (section [2.2.5.4](#)). This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_ImportPackage (  
    @PartitionId uniqueidentifier  
    ,@Id varchar(512)  
    ,@Type uniqueidentifier  
    ,@IsPublished bit  
    ,@LastModifiedTime datetime  
    ,@ContentSize int  
    ,@Content varbinary(max)  
);
```

**@PartitionId:** Term store partition identifier of the term store that the content type belongs to. This parameter MUST NOT be NULL.

**@Id:** The identifier of the content type being imported.

**@Type:** Reserved. Value MUST be "B4AD3A44-D934-4C91-8D1F-463ACEADE443".

**@IsPublished:** Indicates whether the content type is a **published content type**.

**@LastModifiedTime:** The date and time, in Coordinated Universal Time (UTC) format, when the content type being imported was last modified.

**@ContentSize:** The size of the content type package.

**@Content:** The content of the content type package.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.64 **proc\_ECM\_AddChangeLog**

This stored procedure adds an entry to the **ECMChangeLog** table (section [2.2.5.2](#)) to record a change to any item in the term store. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_AddChangeLog (  
    @PartitionId uniqueidentifier  
    ,@ObjectType int  
    ,@ChangeType int  
    ,@GroupUniqueId uniqueidentifier = null  
    ,@TermSetUniqueId uniqueidentifier = null  
    ,@ObjectUniqueId uniqueidentifier = null  
    ,@ObjectId int = null  
    ,@ChangeData nvarchar(max) = null  
);
```

**@PartitionId:** The identifier of the term store partition that the change belongs to. This parameter MUST NOT be NULL.

**@ObjectType:** Specifies the type of object that has changed. MUST be one of the values listed in the description of the **ObjectType** column in the **ECMChangeLog** table. This parameter MUST NOT be NULL.

**@ChangeType:** Specifies the type of change. MUST be one of the values listed in the description of the **ChangeType** column in the **ECMChangeLog** table. This parameter MUST NOT be NULL.

**@GroupUniqueId:** The identifier of the group that the change is scoped to. The value MUST be NULL if the change is not scoped inside a group, such as a language change, permission change, partition data import, partition data deletion, and used term change.

**@TermSetUniqueId:** The identifier of the term set that the change is scoped to. The value MUST be NULL if the change is not scoped inside a term set, such as a language change, permission change, partition data import, partition data deletion, group change, or used term change.

**@ObjectUniqueId:** The identifier of the object that the change is for. The value MUST be NULL if the change is not for a specific object, such as a language change, permission change, partition data import, partition data deletion, or used term change.

**@ObjectId:** The internal identifier (1) of the object that the change is for. This value MUST be NULL if the change is not for a specific object, such as a language change, permission change, partition data import, partition data deletion, or used term change.

**@ChangeData:** Contains the data that pertains to the type of change.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.65 proc\_ECM\_GetChangesForFullListSync

The **proc\_ECM\_GetChangesForFullListSync** stored procedure is called to retrieve details of term store changes occurring after a specified time. The details of changes MUST be associated with any of the following operations: term store import and term store restore. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_GetChangesForFullListSync (  
    @SinceTime datetime = null  
);
```

**@SinceTime:** The date and time value, in UTC, after which the changes that occurred MUST be retrieved. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetChanges.proc\\_ECM\\_GetChanges.Default.ResultSet0](#)

### 3.1.5.66 proc\_ECM\_LogRestore

This stored procedure logs an event that affects all term stores in the database or farm. For example, this should be called on a database restore because it is a database-wide change. This should also be called on farm-wide changes such as a full farm install. If a client calls this stored procedure, it is expected to have to flush all its cached data and retrieve it from the database again. This stored procedure creates an entry in the **ECMChangeLog** table (section [2.2.5.2](#)) with a special partition identifier to indicate that it pertains to all term stores. If the change is not farm wide, an entry will also be added to the **ECMPackage** table (section [2.2.5.4](#)) with the special partition identifier and the UTC time of the modification. If a previous entry exists in the **ECMPackage** table for the special partition identifier, that entry will be updated. To log events local to a particular term store, use the **proc\_ECM\_AddChangeLog** stored procedure (section [3.1.5.64](#)) instead. This stored procedure is defined as follows.

```
PROCEDURE proc_ECM_LogRestore (  
    @IsFullFarm bit = 0  
);
```

**@IsFullFarm:** Indicates whether the change pertains to a full farm install. The **ChangeData** field of the **ECMChangeLog** table will contain a string with value 1 if this bit is set to 1; otherwise, it will contain a string with value 0.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.67 proc\_ECM\_UnpublishAllPackages

This stored procedure marks all the content type packages in the specified term store partition as unpublished by setting the **IsPublished** bit for each package in the **ECMPackage** table (section [2.2.5.4](#)) to 0. The last modified time for each content type package in that partition is also updated to the current UTC time. This stored procedure is defined as follows.

```

PROCEDURE proc_ECM_UnpublishAllPackages (
  @PartitionId uniqueidentifier
);

```

**@PartitionId:** Term store partition identifier. This parameter MUST NOT be NULL.

**Return Values:** An integer that MUST be 0.

**Result Sets:** MUST NOT return any result sets.

### 3.1.5.68 proc\_ECM\_GetChangedTermSets

The **proc\_ECM\_GetChangedTermSets** stored procedure is called to retrieve term sets that have changed since a particular time or that contain terms that have changed since that time. The query MUST return data across all partitions. <14>

```

PROCEDURE proc_ECM_GetChangedTermSets (
  @SinceTime datetime
  ,@PropertyName nvarchar(255) = null
  ,@PropertyValue nvarchar(max) = null
);

```

**@SinceTime:** **@SinceTime:** The date and time value, in UTC, after which the changes that occurred MUST be retrieved. This parameter MUST NOT be NULL.

**@PropertyName:** The value specifying the custom property name with which to search. If this value is not NULL or empty then only term sets that have a custom property defined with the **PropertyName** will be returned. The string matching MUST be case-insensitive.

**@PropertyValue:** A value specifying the custom property value with which to search. The string matching MUST be case-insensitive. If this value is NULL then the custom property value MUST be ignored and the match MUST only be performed on the custom property name provided in the **@PropertyName** parameter (if applicable) and changed after the **@SinceTime** parameter.

**Return Values:** An integer which MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetChangedTermSets.ResultSet0](#)

### 3.1.5.69 proc\_ECM\_SearchTermSetsByProperty

The **proc\_ECM\_SearchTermSetsByProperty** stored procedure is called to retrieve term sets that have a matching property. When **GroupId** is not specified, all term set information is returned, including the group and hierarchy information. When **GroupId** is specified, group information is not returned. This stored procedure is defined as follows. <15>

```

PROCEDURE proc_ECM_SearchTermSetsByProperty (
  @PartitionId uniqueidentifier
  ,@ResultSize int
  ,@PropertyName nvarchar(255)
  ,@PropertyValue nvarchar(max) = null
  ,@GroupId int = null
  ,@MatchOption smallint = 1
);

```

);

**@PartitionId:** Term store partition identifier (1). This parameter MUST NOT be NULL.

**@ResultSize:** A value specifying the maximum number of term sets to be retrieved. This value MUST be 0 or greater.

**@PropertyName:** A value specifying the custom property name with which to search. If this value is NULL then all result sets will have 0 rows. The string matching MUST be case-insensitive.

**@PropertyValue:** A value specifying the custom property value with which to search. The string matching MUST be case-insensitive. If this value is NULL then the custom property value MUST be ignored and the match MUST only be performed on the custom property name provided in the **@PropertyName** parameter.

**@GroupId:** A group internal identifier (1) used to restrict the scope of the search. Only term sets that are members of this specified group MUST be retrieved. If this value is NULL then the term sets MUST be retrieved from all term sets.

**@MatchOption:** A value indicating the type of string match to perform on the custom property value. If this value is 1, the custom property value MUST exactly match the value specified in the **@PropertyValue** parameter. If this value is NOT 1, the custom property value MUST start with the value specified in the **@PropertyValue** parameter. This parameter has no effect if **@PropertyValue** is NULL.

**Return Values:** An integer which MUST be 0.

#### **Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet0](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet1](#)

This stored procedure MUST return a [proc\\_ECM\\_GetGroupByGuid.ResultSet2](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#)

This stored procedure MUST return a [proc\\_ECM\\_GetTermSetByGuid.ResultSet5](#)

### **3.1.5.70 proc\_ECM\_SearchAllTermSetsByProperty**

The **proc\_ECM\_SearchAllTermSetsByProperty** stored procedure is called to retrieve all properties of the term sets that have a matching property from across all partitions. The properties returned MUST be ordered in ascending order of the term set identifier. This stored procedure is defined as follows<16>

```
PROCEDURE proc_ECM_SearchAllTermSetsByProperty (  
    @PropertyName nvarchar(255)  
    , @PropertyValue nvarchar(255) = null  
    , @ResultSize int  
    , @StartId int  
);
```

**@PropertyName:** A value specifying the custom property name with which to search. If this value is NULL then all result sets will have 0 rows. The string matching MUST be case-insensitive

**@PropertyValue:** A value specifying the custom property value with which to search. The string matching MUST be case-insensitive. If this value is NULL then the custom property value MUST be ignored and the match MUST only performed on the custom property name provided in the **@PropertyName** parameter..

**@ResultSize:** A value specifying the maximum number of term sets to be retrieved. This value MUST be 0 or greater.

**@StartId:** A value specifying the minimum ID of the term sets to be retrieved. This value MUST be 0 or greater.

**Return Values:** An integer which MUST be 0.

**Result Sets:**

This stored procedure MUST return a [proc\\_ECM\\_SearchAllTermSetsByProperty.ResultSet0](#)

### 3.1.6 Timer Events

None.

### 3.1.7 Other Local Events

None.

## 3.2 Client Details

### 3.2.1 Abstract Data Model

### 3.2.2 Timers

### 3.2.3 Initialization

### 3.2.4 Higher-Layer Triggered Events

### 3.2.5 Message Processing Events and Sequencing Rules

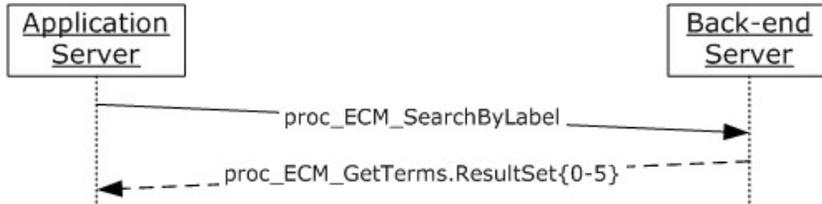
### 3.2.6 Timer Events

### 3.2.7 Other Local Events

## 4 Protocol Examples

### 4.1 Search Terms by a Label

This example describes the request made and the response returned when a user searches terms by a label in a term store.



**Figure 2: Searching terms by a label**

The client searches for terms that starts with "a" by calling stored procedure **proc\_ECM\_SearchByLabel**. Considering the following T-SQL syntax, which displays the parameters used to call the stored procedure:

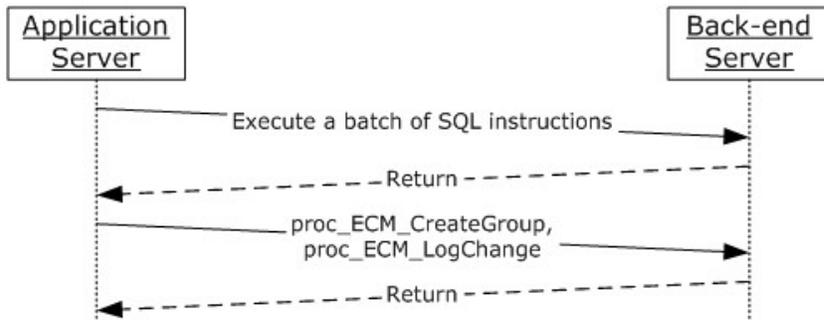
```
exec proc_ECM_SearchByLabel
.....@PartitionId='0C37852B-34D0-418E-91C6-2AC25AF4BE5B'
.....,@Label=N'a'
.....,@Lcid=1033
.....,@DefaultLabelOnly=0
.....,@MatchOption=0
.....,@ResultSize=20
.....,@GetFullPath=1
.....,@DefaultLcid=1033
.....,@IsOptimized=1
```

The server returns the following in order:

- **proc\_ECM\_GetTerms.ResultSet0**, which contains one row for every term that matches the search condition.
- **proc\_ECM\_GetTerms.ResultSet1**, which contains one row for every term label of a term that matches the search condition.
- **proc\_ECM\_GetTerms.ResultSet2**, which contains one row for every term description of a term that matches the search condition.
- **proc\_ECM\_GetTerms.ResultSet3**, which contains one row for every parent term of a term that matches the search condition.
- **proc\_ECM\_GetTerms.ResultSet4**, which contains one row for every custom property of a term that matches the search condition. [<17>](#)
- **proc\_ECM\_GetTerms.ResultSet5**, which contains one row for every immediate child term of a term that matches the search condition.

## 4.2 Create a Group

This example describes the request made and the response returned when a user create a group.



**Figure 3: Creating a Group**

The batch of SQL instructions in step 1 is specified in section 3.1.5. The client creates a group by calling **proc\_ECM\_CreateGroup**, and then **proc\_ECM\_LogChange** as specified in section 3.1.5. Considering the following T-SQL syntax, which displays the parameters used to call the stored procedure:

```
EXEC @retVal=proc_ECM_CreateGroup
.....@PartitionId='0c37852b-34d0-418e-91c6-2ac25af4be5b'
.....,@UniqueId='38d980c0-2ed0-48bc-a863-2446b0a8301c'
.....,@Name=N'A Group Name'
.....,@Description=N'Blah'
.....,@Type='0'
.....,@OldId='-1'.....
```

The stored procedures do not return any result sets.

## 5 Security

### 5.1 Security Considerations for Implementers

Interactions with SQL are susceptible to tampering and other forms of security risks. Implementers are advised to sanitize input parameters for stored procedures prior to invoking the stored procedure.

### 5.2 Index of Security Parameters

Preliminary

## 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® SharePoint® Server 2010
- Microsoft® SharePoint® Server 2013 Preview

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 2.2.4.15:](#) This attribute is not returned by SharePoint Server 2010.

[<2> Section 2.2.4.40:](#) This result set is not returned by SharePoint Server 2010.

[<3> Section 2.2.4.42:](#) This result set is not returned by SharePoint Server 2010.

[<4> Section 2.2.4.43:](#) This attribute is not returned by SharePoint Server 2010.

[<5> Section 2.2.4.44:](#) This attribute is not returned by SharePoint Server 2010.

[<6> Section 2.2.5.10:](#) This is not supported in SharePoint Server 2010.

[<7> Section 2.2.5.14:](#) This is not supported by SharePoint Server 2010.

[<8> Section 3.1.5.5:](#) This parameter is not supported in SharePoint Server 2010.

[<9> Section 3.1.5.6:](#) This parameter is not supported by SharePoint Server 2010.

[<10> Section 3.1.5.11:](#) This parameter is not supported in SharePoint Server 2010.

[<11> Section 3.1.5.20:](#) This parameter is not supported in SharePoint Server 2010.

[<12> Section 3.1.5.56:](#) This is not supported in SharePoint Server 2010.

[<13> Section 3.1.5.58:](#) This is not supported in SharePoint Server 2010.

[<14> Section 3.1.5.68:](#) This stored procedure is not supported in SharePoint Server 2010.

[<15> Section 3.1.5.69:](#) This stored procedure is not supported in SharePoint Server 2010.

[<16> Section 3.1.5.70:](#) This stored procedure is not supported in SharePoint Server 2010.

[<17> Section 4.1:](#) `proc_ECM_GetTerms.ResultSet4` is returned in front of `proc_ECM_GetTerms.ResultSet3` in SharePoint Server 2010.

## 7 Change Tracking

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

Preliminary

## 8 Index

### A

Abstract data model  
    [server](#) 46  
[Applicability](#) 10  
[Attribute groups - overview](#) 42  
[Attributes - overview](#) 42

### C

[Capability negotiation](#) 11  
[Change tracking](#) 111  
Client  
    [overview](#) 46  
[Complex types - overview](#) 42  
[Create a group example](#) 108

### D

Data model - abstract  
    [server](#) 46

### E

[ECMGroup table structure](#) 39  
[ECMLanguage table structure](#) 37  
[ECMMergedTerm table structure](#) 38  
[ECMPackage table structure](#) 35  
[ECMPermission table structure](#) 41  
[ECMServiceSettings table structure](#) 39  
[ECMTerm table structure](#) 36  
ECMTermDescription table structure ([section 2.2.5.13](#) 40, [section 3.1.5.6](#) 52)  
[ECMTermLabel table structure](#) 37  
[ECMTermProperty table structure](#) 38  
[ECMTermSet table structure](#) 34  
[ECMTermSetMembership table structure](#) 40  
[ECMUsedTerms table structure](#) 37  
[Elements - overview](#) 42  
[Errors raised - messages](#) 42  
Events  
    [local - server](#) 106  
    [timer - server](#) 106  
Examples  
    [create a group](#) 108  
    [search terms by label](#) 107

### F

[Fields - vendor-extensible](#) 11

### G

[Glossary](#) 8  
[Groups - overview](#) 42

### H

Higher-layer triggered events

[server](#) 47

### I

[Implementer - security considerations](#) 109  
[Informative references](#) 9  
Initialization  
    [server](#) 47  
[Introduction](#) 8

### L

Local events  
    [server](#) 106

### M

Message processing  
    [server](#) 47  
Messages  
    [attribute groups](#) 42  
    [attributes](#) 42  
    [complex types](#) 42  
    [ECMGroup table structure](#) 39  
    [ECMLanguage table structure](#) 37  
    [ECMMergedTerm table structure](#) 38  
    [ECMPackage table structure](#) 35  
    [ECMPermission table structure](#) 41  
    [ECMServiceSettings table structure](#) 39  
    [ECMTerm table structure](#) 36  
    ECMTermDescription table structure ([section 2.2.5.13](#) 40, [section 3.1.5.6](#) 52)  
    [ECMTermLabel table structure](#) 37  
    [ECMTermProperty table structure](#) 38  
    [ECMTermSet table structure](#) 34  
    [ECMTermSetMembership table structure](#) 40  
    [ECMUsedTerms table structure](#) 37  
    [elements](#) 42  
    [groups](#) 42  
    [namespaces](#) 42  
    [proc\\_ECM\\_DoesPartitionDataExist.ResultSet0 result set](#) 12  
    [proc\\_ECM\\_GetChanges.proc\\_ECM\\_GetChanges.Default.ResultSet0 result set](#) 22  
    [proc\\_ECM\\_GetChangesForListSync.ResultSet0 result set](#) ([section 2.2.4.24](#) 21, [section 3.1.5.32](#) 70)  
    [proc\\_ECM\\_GetChangeTime.ResultSet0 result set](#) 12  
    [proc\\_ECM\\_GetGroupByGuid.ResultSet0 result set](#) 13  
    [proc\\_ECM\\_GetGroupByGuid.ResultSet1 result set](#) 13  
    [proc\\_ECM\\_GetGroupByGuid.ResultSet2 result set](#) 12  
    [proc\\_ECM\\_GetPackage.ResultSet0 result set](#) 14  
    [proc\\_ECM\\_GetPackagesInformation.ResultSet0 result set](#) 14

[proc\\_ECM\\_GetServiceSettings.ResultSet0 result set](#) 14  
[proc\\_ECM\\_GetSessionData.ResultSet0 result set](#) 15  
[proc\\_ECM\\_GetSessionData.ResultSet1 result set](#) 15  
[proc\\_ECM\\_GetSessionData.ResultSet2 result set](#) 15  
[proc\\_ECM\\_GetSessionData.ResultSet5 result set](#) 14  
[proc\\_ECM\\_GetSessionData.ResultSet6 result set](#) 29  
[proc\\_ECM\\_GetSessionData.ResultSet4 result set](#) 29  
[proc\\_ECM\\_GetSortedChildTermsIds.ResultSet0 result set](#) 29  
[proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.LcidNotNull.ResultSet9 result set](#) 30  
[proc\\_ECM\\_GetTerms.ResultSet0 result set](#) 18  
[proc\\_ECM\\_GetTerms.ResultSet1 result set](#) 18  
[proc\\_ECM\\_GetTerms.ResultSet2 result set](#) 17  
[proc\\_ECM\\_GetTerms.ResultSet3 result set](#) 17  
[proc\\_ECM\\_GetTerms.ResultSet4 result set](#) (section 2.2.4.15 16, section 2.2.4.44 32)  
[proc\\_ECM\\_GetTerms.ResultSet5 result set](#) 16  
[proc\\_ECM\\_GetTermSetByGuid.ResultSet3 result set](#) 19  
[proc\\_ECM\\_GetTermSetByGuid.ResultSet4 result set](#) (section 2.2.4.20 19, section 2.2.4.42 30)  
[proc\\_ECM\\_MoveTermSetMembership.ResultSet0 result set](#) 16  
[proc\\_ECM\\_RetrieveApplicationLog.ResultSet0 result set](#) 20  
[proc\\_ECM\\_RetrieveTableNames.ResultSet0 result set](#) 21  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMMerqedTerm.ResultSet0 result set](#) 26  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPackage.ResultSet0 result set](#) 24  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPermission.ResultSet0 result set](#) 28  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMServiceSettings.ResultSet0 result set](#) 27  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTerm.ResultSet0 result set](#) 25  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermDescription.ResultSet0 result set](#) 27  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermLabel.ResultSet0 result set](#) 23  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermProperty.ResultSet0 result set](#) 26  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermSetMembership.ResultSet0 result set](#) 28

[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMUsedTerms.ResultSet0 result set](#) 23  
[raised errors](#) 42  
[simple types](#) 42  
[transport](#) 12  
[XML structures](#) 42

#### Methods

[proc\\_ECM\\_AddApplicationLogEntry](#) 49  
[proc\\_ECM\\_AddChangeLog](#) 102  
[proc\\_ECM\\_AddLanguage](#) 50  
[proc\\_ECM\\_AddTermDescription](#) 50  
[proc\\_ECM\\_AddTermLabel](#) 48  
[proc\\_ECM\\_AddTermProperty](#) 51  
[proc\\_ECM\\_AddTermSetMembership](#) 52  
[proc\\_ECM\\_AddUsedTerm](#) 53  
[proc\\_ECM\\_ClearPartitionData](#) 54  
[proc\\_ECM\\_ClearUsedTerms](#) 100  
[proc\\_ECM\\_CreateGroup](#) 55  
[proc\\_ECM\\_CreatePermission](#) 55  
[proc\\_ECM\\_CreateTerm](#) 56  
[proc\\_ECM\\_CreateTermSet](#) 57  
[proc\\_ECM\\_DeleteGroup](#) 59  
[proc\\_ECM\\_DeleteLanguage](#) 60  
[proc\\_ECM\\_DeletePermission](#) 60  
[proc\\_ECM\\_DeleteServiceSettings](#) 61  
[proc\\_ECM\\_DeleteTerm](#) 62  
[proc\\_ECM\\_DeleteTermDescription](#) 62  
[proc\\_ECM\\_DeleteTermLabel](#) 63  
[proc\\_ECM\\_DeleteTermProperty](#) 63  
[proc\\_ECM\\_DeleteTermSet](#) 63  
[proc\\_ECM\\_DoesPartitionDataExist](#) 64  
[proc\\_ECM\\_GetAllTerms](#) 65  
[proc\\_ECM\\_GetChangedTermSets](#) 104  
[proc\\_ECM\\_GetChanges](#) 65  
[proc\\_ECM\\_GetChangesForFullListSync](#) 103  
[proc\\_ECM\\_GetChangesForListSync](#) 101  
[proc\\_ECM\\_GetChangeTime](#) 66  
[proc\\_ECM\\_GetDescendentTerms](#) 67  
[proc\\_ECM\\_GetGroupByGuid](#) 68  
[proc\\_ECM\\_GetGroups](#) 68  
[proc\\_ECM\\_GetPackage](#) 69  
[proc\\_ECM\\_GetPackagesInformation](#) 69  
[proc\\_ECM\\_GetServiceSettings](#) 70  
[proc\\_ECM\\_GetSessionData](#) 70  
[proc\\_ECM\\_GetSortedChildTermIds](#) 71  
[proc\\_ECM\\_GetTermInTermSetByGuid](#) 71  
[proc\\_ECM\\_GetTerms](#) 72  
[proc\\_ECM\\_GetTermSets](#) 75  
[proc\\_ECM\\_GetTermSetsByLabel](#) 77  
[proc\\_ECM\\_ImportPackage](#) 101  
[proc\\_ECM\\_ImportPartitionData](#) 78  
[proc\\_ECM\\_LogChange](#) 79  
[proc\\_ECM\\_LogRestore](#) 103  
[proc\\_ECM\\_MergeTerm](#) 79  
[proc\\_ECM\\_MoveTermSetMembership](#) 81  
[proc\\_ECM\\_PopulateLabelsForDefaultLanguage](#) 82  
[proc\\_ECM\\_RemoveApplicationLog](#) 83  
[proc\\_ECM\\_RemoveUsedTermSets](#) 83  
[proc\\_ECM\\_RetrieveApplicationLog](#) 83  
[proc\\_ECM\\_RetrieveTableNames](#) 84

[proc ECM SearchAllTermSetsByProperty](#) 105  
[proc ECM SearchByLabel](#) 84  
[proc ECM SearchByProperty](#) 87  
[proc ECM SearchTermSetByName](#) 88  
[proc ECM SearchTermSetsByProperty](#) 104  
[proc ECM SelectPartitionData](#) 88  
[proc ECM SetDefaultLanguage](#) 91  
[proc ECM SetPackage](#) 92  
[proc ECM SetServiceSettings](#) 92  
[proc ECM UnpublishAllPackages](#) 103  
[proc ECM UpdateGroup](#) 93  
[proc ECM UpdatePermission](#) 94  
[proc ECM UpdateTerm](#) 95  
[proc ECM UpdateTermDescription](#) 96  
[proc ECM UpdateTermProperty](#) 96  
[proc ECM UpdateTermSet](#) 97  
[proc ECM UpdateTermSetMembership](#) 99

## N

[Namespaces](#) 42  
[Normative references](#) 9

## O

[Overview \(synopsis\)](#) 10

## P

[Preconditions](#) 10  
[Prerequisites](#) 10  
[proc ECM AddApplicationLogEntry method](#) 49  
[proc ECM AddChangeLog method](#) 102  
[proc ECM AddLanguage method](#) 50  
[proc ECM AddTermDescription method](#) 50  
[proc ECM AddTermLabel method](#) 48  
[proc ECM AddTermProperty method](#) 51  
[proc ECM AddTermSetMembership method](#) 52  
[proc ECM AddUsedTerm method](#) 53  
[proc ECM ClearPartitionData method](#) 54  
[proc ECM ClearUsedTerms method](#) 100  
[proc ECM CreateGroup method](#) 55  
[proc ECM CreatePermission method](#) 55  
[proc ECM CreateTerm method](#) 56  
[proc ECM CreateTermSet method](#) 57  
[proc ECM DeleteGroup method](#) 59  
[proc ECM DeleteLanguage method](#) 60  
[proc ECM DeletePermission method](#) 60  
[proc ECM DeleteServiceSettings method](#) 61  
[proc ECM DeleteTerm method](#) 62  
[proc ECM DeleteTermDescription method](#) 62  
[proc ECM DeleteTermLabel method](#) 63  
[proc ECM DeleteTermProperty method](#) 63  
[proc ECM DeleteTermSet method](#) 63  
[proc ECM DoesPartitionDataExist method](#) 64  
[proc ECM DoesPartitionDataExist.ResultSet0 result set](#) 12  
[proc ECM GetAllTerms method](#) 65  
[proc ECM GetChangedTermSets method](#) 104  
[proc ECM GetChanges method](#) 65  
[proc ECM GetChanges.proc ECM GetChanges.Default.ResultSet0 result set](#) 22

[proc ECM GetChangesForFullListSync method](#) 103  
[proc ECM GetChangesForListSync method](#) 101  
[proc ECM GetChangesForListSync.ResultSet0 result set](#) (section 2.2.4.24 21, section 3.1.5.32 70)  
[proc ECM GetChangeTime method](#) 66  
[proc ECM GetChangeTime.ResultSet0 result set](#) 12  
[proc ECM GetDescendentTerms method](#) 67  
[proc ECM GetGroupByGuid method](#) 68  
[proc ECM GetGroupByGuid.ResultSet0 result set](#) 13  
[proc ECM GetGroupByGuid.ResultSet1 result set](#) 13  
[proc ECM GetGroupByGuid.ResultSet2 result set](#) 12  
[proc ECM GetGroups method](#) 68  
[proc ECM GetPackage method](#) 69  
[proc ECM GetPackage.ResultSet0 result set](#) 14  
[proc ECM GetPackagesInformation method](#) 69  
[proc ECM GetPackagesInformation.ResultSet0 result set](#) 14  
[proc ECM GetServiceSettings method](#) 70  
[proc ECM GetServiceSettings.ResultSet0 result set](#) 14  
[proc ECM GetSessionData method](#) 70  
[proc ECM GetSessionData.ResultSet0 result set](#) 15  
[proc ECM GetSessionData.ResultSet1 result set](#) 15  
[proc ECM GetSessionData.ResultSet2 result set](#) 15  
[proc ECM GetSessionData.ResultSet5 result set](#) 14  
[proc ECM GetSessionData.ResultSet6 result set](#) 29  
[proc ECM GetSessionData.ResultSet4 result set](#) 29  
[proc ECM GetSortedChildTermIds method](#) 71  
[proc ECM GetSortedChildTermsIds.ResultSet0 result set](#) 29  
[proc ECM GetTermInTermSetByGuid method](#) 71  
[proc ECM GetTerms method](#) 72  
[proc ECM GetTerms.proc ECM GetTerms.LcidNotNull.ResultSet9 result set](#) 30  
[proc ECM GetTerms.ResultSet0 result set](#) 18  
[proc ECM GetTerms.ResultSet1 result set](#) 18  
[proc ECM GetTerms.ResultSet2 result set](#) 17  
[proc ECM GetTerms.ResultSet3 result set](#) 17  
[proc ECM GetTerms.ResultSet4 result set](#) (section 2.2.4.15 16, section 2.2.4.44 32)  
[proc ECM GetTerms.ResultSet5 result set](#) 16  
[proc ECM GetTermSetByGuid.ResultSet3 result set](#) 19  
[proc ECM GetTermSetByGuid.ResultSet4 result set](#) (section 2.2.4.20 19, section 2.2.4.42 30)  
[proc ECM GetTermSets method](#) 75  
[proc ECM GetTermSetsByLabel method](#) 77  
[proc ECM ImportPackage method](#) 101  
[proc ECM ImportPartitionData method](#) 78  
[proc ECM LogChange method](#) 79  
[proc ECM LogRestore method](#) 103  
[proc ECM MergeTerm method](#) 79  
[proc ECM MoveTermSetMembership method](#) 81  
[proc ECM MoveTermSetMembership.ResultSet0 result set](#) 16  
[proc ECM PopulateLabelsForDefaultLanguage method](#) 82

[proc\\_ECM\\_RemoveApplicationLog\\_method](#) 83  
[proc\\_ECM\\_RemoveUsedTermSets\\_method](#) 83  
[proc\\_ECM\\_RetrieveApplicationLog\\_method](#) 83  
[proc\\_ECM\\_RetrieveApplicationLog.ResultSet0\\_result\\_set](#) 20  
[proc\\_ECM\\_RetrieveTableNames\\_method](#) 84  
[proc\\_ECM\\_RetrieveTableNames.ResultSet0\\_result\\_set](#) 21  
[proc\\_ECM\\_SearchAllTermSetsByProperty\\_method](#) 105  
[proc\\_ECM\\_SearchByLabel\\_method](#) 84  
[proc\\_ECM\\_SearchByProperty\\_method](#) 87  
[proc\\_ECM\\_SearchTermSetByName\\_method](#) 88  
[proc\\_ECM\\_SearchTermSetsByProperty\\_method](#) 104  
[proc\\_ECM\\_SelectPartitionData\\_method](#) 88  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMMergedTerm.ResultSet0\\_result\\_set](#) 26  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPackage.ResultSet0\\_result\\_set](#) 24  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPermission.ResultSet0\\_result\\_set](#) 28  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMServiceSettings.ResultSet0\\_result\\_set](#) 27  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTerm.ResultSet0\\_result\\_set](#) 25  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermDescription.ResultSet0\\_result\\_set](#) 27  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermLabel.ResultSet0\\_result\\_set](#) 23  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermProperty.ResultSet0\\_result\\_set](#) 26  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermSetMembership.ResultSet0\\_result\\_set](#) 28  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMUsedTerms.ResultSet0\\_result\\_set](#) 23  
[proc\\_ECM\\_SetDefaultLanguage\\_method](#) 91  
[proc\\_ECM\\_SetPackage\\_method](#) 92  
[proc\\_ECM\\_SetServiceSettings\\_method](#) 92  
[proc\\_ECM\\_UnpublishAllPackages\\_method](#) 103  
[proc\\_ECM\\_UpdateGroup\\_method](#) 93  
[proc\\_ECM\\_UpdatePermission\\_method](#) 94  
[proc\\_ECM\\_UpdateTerm\\_method](#) 95  
[proc\\_ECM\\_UpdateTermDescription\\_method](#) 96  
[proc\\_ECM\\_UpdateTermProperty\\_method](#) 96  
[proc\\_ECM\\_UpdateTermSet\\_method](#) 97  
[proc\\_ECM\\_UpdateTermSetMembership\\_method](#) 99  
[Product behavior](#) 110

## R

[Raised errors - messages](#) 42  
[References](#) 9  
   [informative](#) 9  
   [normative](#) 9  
[Relationship to other protocols](#) 10  
[Result sets - messages](#)  
   [proc\\_ECM\\_DoesPartitionDataExist.ResultSet0](#) 12

[proc\\_ECM\\_GetChanges.proc\\_ECM\\_GetChanges.Default.ResultSet0](#) 22  
[proc\\_ECM\\_GetChangesForListSync.ResultSet0](#) (section 2.2.4.24 21, section 3.1.5.32 70)  
[proc\\_ECM\\_GetChangeTime.ResultSet0](#) 12  
[proc\\_ECM\\_GetGroupByGuid.ResultSet0](#) 13  
[proc\\_ECM\\_GetGroupByGuid.ResultSet1](#) 13  
[proc\\_ECM\\_GetGroupByGuid.ResultSet2](#) 12  
[proc\\_ECM\\_GetPackage.ResultSet0](#) 14  
[proc\\_ECM\\_GetPackagesInformation.ResultSet0](#) 14  
[proc\\_ECM\\_GetServiceSettings.ResultSet0](#) 14  
[proc\\_ECM\\_GetSessionData.ResultSet0](#) 15  
[proc\\_ECM\\_GetSessionData.ResultSet1](#) 15  
[proc\\_ECM\\_GetSessionData.ResultSet2](#) 15  
[proc\\_ECM\\_GetSessionData.ResultSet5](#) 14  
[proc\\_ECM\\_GetTerms.ResultSet0](#) 18  
[proc\\_ECM\\_GetTerms.ResultSet1](#) 18  
[proc\\_ECM\\_GetTerms.ResultSet2](#) 17  
[proc\\_ECM\\_GetTerms.ResultSet3](#) 17  
[proc\\_ECM\\_GetTerms.ResultSet4](#) (section 2.2.4.15 16, section 2.2.4.44 32)  
[proc\\_ECM\\_GetTerms.ResultSet5](#) 16  
[proc\\_ECM\\_GetTermSetByGuid.ResultSet3](#) 19  
[proc\\_ECM\\_GetTermSetByGuid.ResultSet4](#) (section 2.2.4.20 19, section 2.2.4.42 30)  
[proc\\_ECM\\_MoveTermSetMembership.ResultSet0](#) 16  
[proc\\_ECM\\_RetrieveApplicationLog.ResultSet0](#) 20  
[proc\\_ECM\\_RetrieveTableNames.ResultSet0](#) 21  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMMergedTerm.ResultSet0](#) 26  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPackage.ResultSet0](#) 24  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMPermission.ResultSet0](#) 28  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMServiceSettings.ResultSet0](#) 27  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTerm.ResultSet0](#) 25  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermDescription.ResultSet0](#) 27  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermLabel.ResultSet0](#) 23  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermProperty.ResultSet0](#) 26  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMTermSetMembership.ResultSet0](#) 28  
[proc\\_ECM\\_SelectPartitionData.proc\\_ECM\\_SelectPartitionData.ECMUsedTerms.ResultSet0](#) 23  
[Result sets – messages](#)  
   [proc\\_ECM\\_GetSessionData.ResultSet6](#) 29  
   [proc\\_ECM\\_GetSessionData.ResultSet4](#) 29  
   [proc\\_ECM\\_GetSortedChildTermsIds.ResultSet0](#) 29  
   [proc\\_ECM\\_GetTerms.proc\\_ECM\\_GetTerms.LcidNotNull.ResultSet9](#) 30

## S

[Search terms by label example](#) 107

Security

- [implementer considerations](#) 109

Sequencing rules

- [server](#) 47

Server

- [abstract data model](#) 46
- [higher-layer triggered events](#) 47
- [initialization](#) 47
- [local events](#) 106
- [message processing](#) 47
- [overview](#) 46
- [proc ECM AddApplicationLogEntry method](#) 49
- [proc ECM AddChangeLog method](#) 102
- [proc ECM AddLanguage method](#) 50
- [proc ECM AddTermDescription method](#) 50
- [proc ECM AddTermLabel method](#) 48
- [proc ECM AddTermProperty method](#) 51
- [proc ECM AddTermSetMembership method](#) 52
- [proc ECM AddUsedTerm method](#) 53
- [proc ECM ClearPartitionData method](#) 54
- [proc ECM ClearUsedTerms method](#) 100
- [proc ECM CreateGroup method](#) 55
- [proc ECM CreatePermission method](#) 55
- [proc ECM CreateTerm method](#) 56
- [proc ECM CreateTermSet method](#) 57
- [proc ECM DeleteGroup method](#) 59
- [proc ECM DeleteLanguage method](#) 60
- [proc ECM DeletePermission method](#) 60
- [proc ECM DeleteServiceSettings method](#) 61
- [proc ECM DeleteTerm method](#) 62
- [proc ECM DeleteTermDescription method](#) 62
- [proc ECM DeleteTermLabel method](#) 63
- [proc ECM DeleteTermProperty method](#) 63
- [proc ECM DeleteTermSet method](#) 63
- [proc ECM DoesPartitionDataExist method](#) 64
- [proc ECM GetAllTerms method](#) 65
- [proc ECM GetChangedTermSets method](#) 104
- [proc ECM GetChanges method](#) 65
- [proc ECM GetChangesForFullListSync method](#) 103
- [proc ECM GetChangesForListSync method](#) 101
- [proc ECM GetChangeTime method](#) 66
- [proc ECM GetDescendentTerms method](#) 67
- [proc ECM GetGroupByGuid method](#) 68
- [proc ECM GetGroups method](#) 68
- [proc ECM GetPackage method](#) 69
- [proc ECM GetPackagesInformation method](#) 69
- [proc ECM GetServiceSettings method](#) 70
- [proc ECM GetSessionData method](#) 70
- [proc ECM GetSortedChildTermIds method](#) 71
- [proc ECM GetTermInTermSetByGuid method](#) 71
- [proc ECM GetTerms method](#) 72
- [proc ECM GetTermSets method](#) 75
- [proc ECM GetTermSetsByLabel method](#) 77
- [proc ECM ImportPackage method](#) 101
- [proc ECM ImportPartitionData method](#) 78
- [proc ECM LogChange method](#) 79
- [proc ECM LogRestore method](#) 103
- [proc ECM MergeTerm method](#) 79
- [proc ECM MoveTermSetMembership method](#) 81

- [proc ECM PopulateLabelsForDefaultLanguage method](#) 82
- [proc ECM RemoveApplicationLog method](#) 83
- [proc ECM RemoveUsedTermSets method](#) 83
- [proc ECM RetrieveApplicationLog method](#) 83
- [proc ECM RetrieveTableNames method](#) 84
- [proc ECM SearchAllTermSetsByProperty method](#) 105
- [proc ECM SearchByLabel method](#) 84
- [proc ECM SearchByProperty method](#) 87
- [proc ECM SearchTermSetByName method](#) 88
- [proc ECM SearchTermSetsByProperty method](#) 104
- [proc ECM SelectPartitionData method](#) 88
- [proc ECM SetDefaultLanguage method](#) 91
- [proc ECM SetPackage method](#) 92
- [proc ECM SetServiceSettings method](#) 92
- [proc ECM UnpublishAllPackages method](#) 103
- [proc ECM UpdateGroup method](#) 93
- [proc ECM UpdatePermission method](#) 94
- [proc ECM UpdateTerm method](#) 95
- [proc ECM UpdateTermDescription method](#) 96
- [proc ECM UpdateTermProperty method](#) 96
- [proc ECM UpdateTermSet method](#) 97
- [proc ECM UpdateTermSetMembership method](#) 99
- [sequencing rules](#) 47
- [timer events](#) 106
- [timers](#) 47
- [Simple types - overview](#) 42
- [Standards assignments](#) 11

Structures

- [XML](#) 42

## T

Table structures

- [ECMGroup](#) 39
- [ECMLanguage](#) 37
- [ECMMergedTerm](#) 38
- [ECMPackage](#) 35
- [ECMPermission](#) 41
- [ECMServiceSettings](#) 39
- [ECMTerm](#) 36
- [ECMTermDescription](#) ([section 2.2.5.13](#) 40, [section 3.1.5.6](#) 52)
- [ECMTermLabel](#) 37
- [ECMTermProperty](#) 38
- [ECMTermSet](#) 34
- [ECMTermSetMembership](#) 40
- [ECMUsedTerms](#) 37

Timer events

- [server](#) 106

Timers

- [server](#) 47

- [Tracking changes](#) 111
- [Transport](#) 12

Triggered events - higher-layer

- [server](#) 47

Types

- [complex](#) 42
- [simple](#) 42

## **V**

[Vendor-extensible fields](#) 11  
[Versioning](#) 11

## **X**

[XML structures](#) 42

Preliminary