[MS-UPSCDAP]:
User Profile Synchronization (UPS): Configuration Database Data Access Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.

- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.

- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.

- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.

- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.

- **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
08/14/2009	0.1	Major	First Release.
09/25/2009	0.2	Minor	Updated the technical content.
11/06/2009	0.2.1	Editorial	Revised and edited the technical content.
12/18/2009	0.3	Minor	Updated the technical content.
01/29/2010	0.3.1	Editorial	Revised and edited the technical content.
03/12/2010	1.0	Major	Updated and revised the technical content.
04/23/2010	1.0.1	Editorial	Revised and edited the technical content.
06/04/2010	1.0.2	Editorial	Revised and edited the technical content.
07/16/2010	2.0	Major	Significantly changed the technical content.
08/27/2010	3.0	Major	Significantly changed the technical content.
10/08/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
03/25/2011	3.0	No change	No changes to the meaning, language, or formatting of the technical content.
05/06/2011	4.0	Major	Significantly changed the technical content.
06/17/2011	4.1	Minor	Clarified the meaning of the technical content.
09/23/2011	4.1	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
12/16/2011	5.0	Major	Significantly changed the technical content.
03/30/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
07/12/2012	5.0	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

1 Introduction

This document specifies the communication between the User Profile Synchronization Configuration Service acting as a client and the User Profile Synchronization Configuration Database acting as a server.

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]:

attribute
domain name
globally unique identifier (GUID)
principal
security identifier (SID)
stored procedure
UTC (Coordinated Universal Time)

The following terms are specific to this document:

Extensible Object Markup Language (XOML): A declarative XML-based language that is used to initialize structured values and objects and, in particular, to describe **workflows**.

management agent (MA): An object that translates the operation of the **synchronization engine** into the format that a connected data source understands.

metaverse (MV): A storage area of the **synchronization engine** that contains the aggregated information from multiple connected data sources.

multivalued attribute: An **attribute** that can contain multiple values of the same data type.

object resource: A User Profile Synchronization (UPS) database representation of an entity. For example, a User **object resource** represents all User **attribute** values.

request: An object resource that embodies a read, create, delete, or modify Web service message received by the server to operate on a target resource.

result set: A set of records that results from running a **stored procedure** or query, or from applying a filter. The structure and content of the data in a **result set** varies according to the implementation.

synchronization engine: The core processing component that creates an integrated view of objects that are stored in multiple data sources and that manages information in those data sources. This integrated view is determined by the information that is retrieved from the data sources and a set of rules that determines how to process this information.

T-SQL (Transact-Structured Query Language): Statements that applications use to communicate with an instance of SQL Server.

target object resource: An **object resource** on which a **request object resource** will operate.

workflow: (1) The automation of business processes, where business documents and tasks are passed automatically from one user to another for action, according to a set sequence. (2) A structured modular component that enables the automated movement of documents or items through a specific sequence of actions or tasks that are related to built-in or user-defined business processes.

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 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. 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.

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

[MS-UPSCDS] Microsoft Corporation, "User Profile Synchronization (UPS): Configuration Data Structure".

[MS-UPSCP] Microsoft Corporation, "User Profile Synchronization (UPS): Configuration Protocol Extensions".

[MS-UPSDBDAP] Microsoft Corporation, "User Profile Synchronization (UPS): Database Data Access Protocol Specification".

[MS-UPSMODS] Microsoft Corporation, "User Profile Synchronization (UPS): Management Objects Data Structure".

[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-2ED] World Wide Web Consortium, "Namespaces in XML 1.0 (Second Edition)", August 2006, http://www.w3.org/TR/2006/REC-xml-names-20060816/

[XMLSCHEMA1] Thompson, H.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[XMLSCHEMA2] Biron, P.V., Ed., and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

1.2.2 Informative References

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

[MSDN-RegEx] Microsoft Corporation, "Regular Expression Language Elements", http://msdn.microsoft.com/en-us/library/az24scfc(VS.80).aspx

1.3 Overview

This protocol specifies the communication between the User Profile Synchronization (UPS) Service and the User Profile Synchronization Configuration Database Server. This protocol is used by the Synchronization Service to persistently store and maintain configuration data that is subsequently used by the synchronization service while it synchronizes data between connected data stores. This client-to-server protocol specifies **stored procedures** that are implemented by the Configuration Database Server and invoked by the Synchronization Service. It is specified in terms of **Transact-SQL** stored procedures that run at the Database server and are invoked by the Synchronization Service. It uses the MS-Tabular Data Stream Protocol [MS-TDS] as its transport.

As described in section 1.3 of [MS-UPSCDS] the purpose of the synchronization service is to synchronize user identity data between connected data sources. The detail of how this synchronization is performed is controlled through the configuration of the synchronization service; in particular, through the configuration of **management agents** and the **metaverse**.

Configuration for the **synchronization engine** can be stored in a configuration database. This protocol defines a mechanism for the storage and retrieval of the synchronization configuration state in a database. User Profile Synchronization Configuration Overview illustrates the various components and protocols that are involved in User Profile Synchronization, including where this protocol fits.

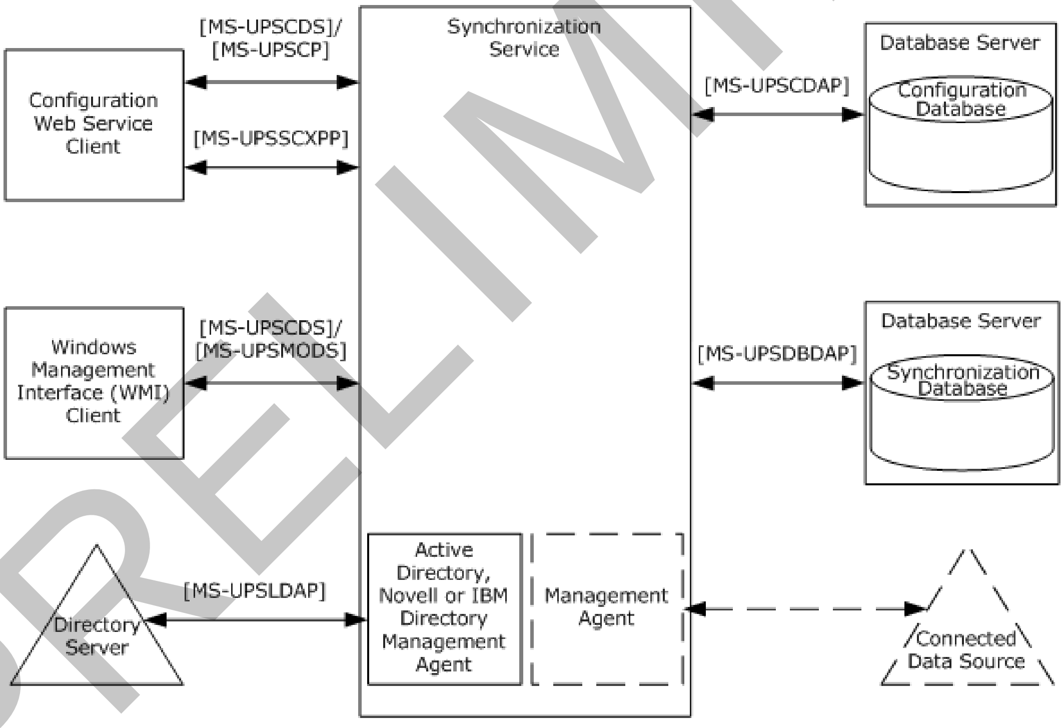

Figure 1: User Profile Synchronization Configuration Overview

For more information on other protocols involved in User Profile Synchronization, see [MS-UPSCDS], [MS-UPSCP], [MS-UPSCXPP], [MS-UPSMODS], [MS-UPSLDAP], and [MS-UPSDBDAP].

The main task for the protocol is to configure the synchronization engine management agent data and metaverse data to enable synchronization activities between different connected data stores. For more information about the management agent configuration data and the metaverse configuration data, see [MS-UPSCDS] and [MS-UPSDBDAP]. A second task of this protocol is to configure the set of users that can operate in the role of configuration administrator, and thus, are authorized to make changes to the synchronization configuration.

This protocol defines the following kinds of objects: Request, User, ma-data, mv-data, Management Policy Rule, and Set. User, ma-data, and mv-data represent the configuration of the synchronization server. **Request** objects represent a client request to change a configuration object. Set is a collection of configuration objects. Management Policy Rule manages the permissions of Create, Read, Update, or Delete operations on target resources. In addition, Management Policy Rule defines workflows that execute during authentication, authorization, and action steps during the processing of a request. During the processing of a request, additional requests, each called Child Request, can be generated.

Objects are represented as a collection of **attributes**. A Binding defines the set of attributes that are associated with an object type. Attributes have one of seven underlying data types: Boolean, binary, integer, string, localized string, text, and localized text. Each attribute has a value and metadata stored with it. The metadata consists of the Object Type Key, Attribute Type Key, created Cause (who or what is creating this attribute), created Time, deleted Cause (who or what is deleting this attribute), and deleted Time. Attributes can be single-valued or multivalued. A **multivalued attribute** is an attribute than can appear multiple times for a given **object resource** instance. A single-valued attribute will appear only once for a given object resource.

This protocol defines a set of primitive operations for adding or removing object attributes. For each of the seven attribute data types, the protocol defines an AddObjectValue operation and a DeleteObjectValue operation for managing attributes that are specific to that data type. Deleting an attribute does not physically delete it, but instead marks the attribute as deleted by setting the deleted Time and deleted Cause. The protocol also defines a set of stored procedures for processing change requests, which includes creating requests, authorizing requests, and committing requests.

Changes to User, ma-data, and mv-data objects are done by first calling a stored procedure for creating and persisting a Request object that represents the configuration change that is to be made. Persisting the Request object provides an audit trail of configuration changes. Once a Request object has been created and persisted in the configuration database, the Request object is evaluated and processed through a sequence of calls to the stored procedures defined by this protocol.

The general sequence of calls for making an update to an object is:

1. CreateRequest: The Request object is created and persisted in the configuration database.

2. EvaluateRequest: The Request object is evaluated to ensure that the user that submitted the Request has the permission to perform the operations specified by the request. In addition, it finds the objects affected by this request and returns their identifiers and the **workflows** definitions that are required to be run on them.

3. PreUpdateProcess: Performs schema validation on the data that passed part of the request.

4. Update Target Object Attributes: The client makes a sequence of calls to the AddObjectValue* or DeleteObjectValue* stored procedures to update attributes of the target object.

5. PostUpdateProcess: The client calls the PostUpdateProcess stored procedure to add newly created configuration objects to their corresponding Sets.

The client side of this protocol is triggered by the necessity, in the synchronization service, to make changes to the synchronization configuration information. This can, in turn, be triggered through operations against an external configuration interface or protocol of the synchronization service. As illustrated in User Profile Synchronization Configuration Overview, an example of one such external protocol is User Profile Synchronization (UPS): Configuration Protocol Extensions [MS-UPSCP].

1.4 Relationship to Other Protocols

The User Profile Synchronization Configuration Data Access protocol depends upon the T-SQL protocol. The T-SQL protocol depends on the Tabular Data Stream (TDS) protocol which depends upon a network transport connection being established prior to a TDS conversation occurring (the choice of transport protocol is not important to TDS). TDS depends on Transport Layer Security (TLS)/Secure Socket Layer (SSL) for network channel encryption. Although the TDS protocol depends on TLS/SSL to encrypt data transmission, the negotiation of the encryption setting between the client and server and the initial TLS/SSL handshake are handled in the TDS layer.

The following diagram shows the transport stack that is used by this protocol.

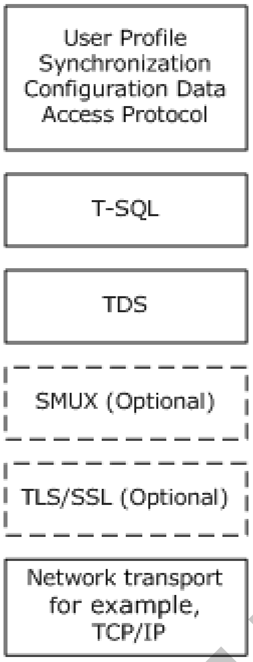
User Profile Synchronization Configuration Data Access Protocol
T-SQL
TDS
SMUX (Optional)
TLS/SSL (Optional)
Network transport for example, TCP/IP

Figure 2: UPS Configuration Data Access protocol in relationship to other protocols

1.5 Prerequisites/Preconditions

The procedures described by this protocol operate between a User Profile Synchronization Configuration Service and a User Profile Synchronization Configuration Database server. The User Profile Synchronization Configuration Service Server is expected to know the location and connection information for the User Profile Synchronization Configuration Database.

This protocol requires that the User Profile Synchronization Configuration Service has permissions to call the stored procedures in the User Profile Synchronization Configuration Database.

1.6 Applicability Statement

This protocol is only applicable to the User Profile Synchronization Configuration Service when communicating with the User Profile Synchronization Configuration Database for user and group administration operations.

1.7 Versioning and Capability Negotiation

In order for the stored procedures to be called correctly, the version of the data structures or stored procedures in the database is defined to match the User Profile Synchronization Configuration Service as described in this document. If the stored procedures do not provide the same signature as expected, the results of the call are indeterminate.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

MS-Tabular Data Stream Protocol [MS-TDS] is the transport protocol used to call the stored procedures, query SQL tables, return result codes, and return **result sets**.

2.2 Common Data Types

The following are common data types used in conjunction with this protocol.

2.2.1 Simple Data Types and Enumerations

2.2.1.1 Object Identifier

An Object Identifier is a **GUID**, as defined in [MS-DTYP] section 2.3.2, used to uniquely identify an instance of an object resource. Object Identifiers are immutable.

2.2.1.2 Object Type Key

An Object Type Key is a smallint value that represents an object resource type. All valid values that are used in this protocol are defined in the table in section 2.2.1.3.

2.2.1.3 Object Type Name

An Object Type Name is a Unicode string value that represents the name of an object resource type. The valid values that are used in this protocol are defined in the following table.

Object Type Key	Object Type Name	Meaning
1	ObjectTypeDescription	Object Type Description.
2	AttributeTypeDescription	Attribute Type Description.
3	BindingDescription	Binding Description.
21	ManagementPolicyRule	Management Policy Rule.
24	Person	User.
26	Request	Request.
28	Set	Set.
34	WorkflowDefinition	Workflow Definition.
35	WorkflowInstance	Workflow Instance.
36	ma-data	Management agent data.
37	mv-data	Metaverse data.

2.2.1.4 Attribute Data Type

The Attribute Data Type is a Unicode string that represents the data type of an attribute of an object resource. The valid values that are used in this protocol are listed in the following table.

Value	SQL Data type
Binary	varbinary(max)
Boolean	Bit
DateTime	Datetime
Integer	Bigint
Reference	Uniqueidentifier
String	nvarchar(488)
Text	nvarchar(MAX)

2.2.1.5 Attribute Type Key

An Attribute Type Key is a smallint value that identifies an attribute type. The valid values that are used by this protocol are defined in the table in section 2.2.1.6.

2.2.1.6 Attribute Type Name

An Attribute Type Name is a Unicode string value that represents the name of an attribute type. The valid values that are used in this protocol are defined in the following tables. Each table lists the attributes that are defined for each Object Resource that is supported by this protocol. For each attribute, the table shows the Attribute Type Key, the Attribute Data Type, and the Meaning. The first table shows the common attributes that exist on all of these resources.

2.2.1.6.1 Common Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
0	ObjectID	Reference	No	Object Identifier.
1	ObjectType	String	No	Object Type Name.
4	CreatedTime	DateTime	No	The time, in **UTC**, when the Resource was created.
5	Creator	Reference	No	The Object Identifier of the Resource that created this Resource.
9	Description	String	No	A human-readable description for the Resource.
11	DisplayName	String	No	A human-readable name for the Resource.

2.2.1.6.2 ObjectTypeDescription Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
19	Name	String	No	Object Type Name.

2.2.1.6.3 AttributeTypeDescription Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
7	DataType	String	No	The Attribute Data Type.
14	IntegerMaximum	Integer	No	The maximum permitted value of the attribute.
15	IntegerMinimum	Integer	No	The minimum permitted value of the attribute.
18	Multivalued	Boolean	No	1, if the attribute is a Multivalued attribute, 0 if not.
19	Name	String	No	Attribute Type Name.
22	StringRegex	String	No	A Regular Expression for the valid values of the attribute. More details about the Regular Expression are contained in [MSDN-RegEx].

2.2.1.6.4 BindingDescription Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
2	BoundAttributeType	Reference	No	The Object Identifier of the Attribute Type Description of this binding.
3	BoundObjectType	Reference	No	The Object Identifier of the Object Type Description of this binding.
14	IntegerMaximum	Integer	No	The maximum permitted value of the attribute.
15	IntegerMinimum	Integer	No	The minimum permitted value of the attribute.
22	StringRegex	String	No	A Regular Expression for the valid values of the attribute. More details about the Regular Expression are contained in [MSDN-RegEx].

2.2.1.6.5 MA-Data Attributes

The description for each attribute is contained in [MS-UPSCDS] section 2.2 under the XML element whose name matches the attribute's Attribute Type Name. The value stored in that attribute is the XML element content of the XML element.

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Name of Corresponding Element in ma-data
189	SyncConfig-id	String	No	SyncConfig-id
190	SyncConfig-category	String	No	SyncConfig-category
191	SyncConfig-sub-type	String	No	SyncConfig-sub-type
192	SyncConfig-ma-listname	String	No	SyncConfig-ma-listname
193	SyncConfig-ma-companyname	String	No	SyncConfig-ma-companyname
194	SyncConfig-creation-time	String	No	SyncConfig-creation-time
195	SyncConfig-last-modification-time	String	No	SyncConfig-last-modification-time
196	SyncConfig-format-version	Integer	No	SyncConfig-format-version
197	SyncConfig-version	Integer	No	SyncConfig-version
198	SyncConfig-internal-version	Integer	No	SyncConfig-internal-version
199	SyncConfig-schema	Text	No	SyncConfig-schema
200	SyncConfig-attribute-inclusion	Text	No	SyncConfig-attribute-inclusion
201	SyncConfig-stay-disconnector	Text	No	SyncConfig-stay-disconnector
202	SyncConfig-join	Text	No	SyncConfig-join
203	SyncConfig-projection	Text	No	SyncConfig-projection
204	SyncConfig-export-attribute-flow	Text	No	SyncConfig-export-attribute-flow
205	SyncConfig-provisioning-cleanup	Text	No	SyncConfig-provisioning-cleanup
206	SyncConfig-provisioning-cleanup-type	String	No	SyncConfig-provisioning-cleanup-type
207	SyncConfig-extension	Text	No	SyncConfig-extension
208	SyncConfig-controller-configuration	Text	No	SyncConfig-controller-configuration
209	SyncConfig-password-	Integer	No	SyncConfig-password-sync-

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Name of Corresponding Element in ma-data
	sync-allowed			allowed
210	SyncConfig-ma-ui-settings	Text	No	SyncConfig-ma-ui-settings
211	SyncConfig-private-configuration	Text	No	SyncConfig-private-configuration
212	SyncConfig-encrypted-attributes	Text	No	SyncConfig-encrypted-attributes
213	SyncConfig-ma-partition-data	Text	Yes	SyncConfig-ma-partition-data
214	SyncConfig-ma-run-data	Text	Yes	SyncConfig-ma-run-data
219	SyncConfig-capabilities-mask	Integer	No	SyncConfig-capabilities-mask
220	SyncConfig-export-type	Integer	No	SyncConfig-export-type
221	SyncConfig-dn-construction	String	No	SyncConfig-dn-construction
222	SyncConfig-password-sync	String	No	SyncConfig-password-sync
223	SyncConfig-component_mappings	String	No	SyncConfig-component_mappings
225	SyncConfig-refresh-schema	Integer	No	SyncConfig-refresh-schema

2.2.1.6.6 Management Policy Rule Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
27	ActionParameter	String	Yes	The Attribute Type Names that the Management Policy Rule applies to. A value of * means it applies to all Attribute Type Names.
28	ActionType	String	Yes	Rule Action Type that this Management Policy Rule applies to.
29	ActionWorkflowDefinition	Reference	Yes	The Object Identifier of a Workflow Definition.
77	Disabled	Boolean	No	1 if this Management Policy Rule is disabled; otherwise, 0.
108	GrantRight	Boolean	No	Specifies whether this Management Policy Rule grants

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
				permission to request the operation in this rule.
150	PrincipalSet	Reference	No	The Object Identifier of a Set that contains Persons that can perform the action.
165	ResourceCurrentSet	Reference	No	The Object Identifier of a Set that a Resource MUST be a member of prior to a Request being executed in order for the Management Policy Rule to apply.
167	ResourceFinalSet	Reference	No	The Object Identifier of a Set that a Resource MUST be a member of after a Request is executed in order for the Management Policy Rule to apply.

2.2.1.6.7 MV-Data Attributes

The description for each attribute is contained in [MS-UPSCDS] section 2.3 under the XML element whose name matches the attribute's Attribute Type Name value in the column Name of Corresponding Element in mv-data. The value stored in that attribute is the XML element content of the XML element.

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Name of Corresponding Element in mv-data
196	SyncConfig-format-version	Integer	No	SyncConfig-format-version
197	SyncConfig-version	Integer	No	SyncConfig-version
199	SyncConfig-schema	Text	No	SyncConfig-schema
207	SyncConfig-extension	Text	No	SyncConfig-extension
215	SyncConfig-import-attribute-flow	Text	No	SyncConfig-import-attribute-flow
216	SyncConfig-mv-deletion	Text	No	SyncConfig-mv-deletion
217	SyncConfig-provisioning	Text	No	SyncConfig-provisioning
218	SyncConfig-provisioning-type	String	No	SyncConfig-provisioning-type
222	SyncConfig-password-sync	String	No	SyncConfig-password-sync
224	SyncConfig-password-change-history-size	Integer	No	SyncConfig-password-change-history-size

2.2.1.6.8 Person Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
25	AccountName	String	No	The account name corresponds to the value of the samAccountName attribute of a User object class in Active Directory, as defined in [MS-ADA3] section 2.222.
26	Domain	String	No	The **domain name** of the user logon domain.
97	FirstName	String	No	The first name of the user.
120	LastName	String	No	The last name of the user.
126	MailNickname	String	No	The email alias of the user.
136	ObjectSID	Binary	No	The **security identifier (SID)**, as defined in [MS-DTYP] section 2.4.2, of the user. The SID is a unique value that is used to identify the user as a security **principal**.

2.2.1.6.9 Request Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
30	ActionWorkflowInstance	Reference	Yes	The Object Identifier of the Workflow Instance that is executed during the processing of the Request.
60	CommittedTime	DateTime	No	The time, in UTC, at which the Request Status became final.
127	ManagementPolicy	Reference	Yes	The Object Identifier of the Management Policy Rules that is triggered by the Request.
140	Operation	String	No	Request Operation.
161	RequestParameter	Text	Yes	Request Parameter.
163	RequestStatus	String	No	Request Status.
180	SetTransition	Text	Yes	The Set Transition element that contains the Object Identifier of a Resource that underwent a Set Transition during processing of this request.
227	Target	Reference	No	The Target Object Identifier.
228	TargetObjectType	String	No	The Object Type Name of the Resource identified by the Target attribute of this request.

2.2.1.6.10 Set Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
24	ComputedMember	Reference	Yes	The Object Identifiers of the Resources that are in the Set as a result of satisfying the predicate defined in the Filter attribute of this Set, in addition to the Object Identifiers on the ExplicitMember attribute.
95	ExplicitMember	Reference	Yes	The Object Identifiers of the Resources that have been explicitly added as members of the Set.
96	Filter	Text	No	A predicate that defines a subset of the Resources. The format of the Filter is defined in section 2.2.3.10 in [MS-UPSCP].
229	Temporal	Boolean	No	1, if the Set Filter attribute matches the Resources based on attributes of the Resource whose Attribute Data Type is datetime; otherwise, 0. This MUST be 0.

2.2.1.6.11 Workflow Definition Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
162	RequestPhase	String	No	This MUST be "Action".
173	RunOnPolicyUpdate	Boolean	No	1, if the workflow is applied to existing members of a ResourceCurrentSet in a Management Policy Rule that references this workflow when the Management Policy Rule is created or enabled; otherwise, 0. This MUST be 0.
243	**XOML**	Text	No	This MUST be the value of Workflow XOML.

2.2.1.6.12 Workflow Instance Attributes

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
158	Request	Reference	No	The Object Identifier of the Request that created this Workflow Instance.
160	Requestor	Reference	No	The Object Identifier of the Person or the Request that created this Workflow Instance.

Attribute Type Key	Attribute Type Name	Attribute Data Type	Multivalued	Meaning
227	Target	Reference	No	Target Object Identifier.
240	WorkflowDefinition	Reference	No	The Object Identifier of the Workflow Definition that is associated with the workflow instance.
241	WorkflowStatus	String	No	Workflow Instance Status.
242	WorkflowStatusDetail	Text	Yes	This attribute is used to store client exception information about the execution of the workflow.

2.2.1.7 Time

Time is a SQL Server datetime that specifies, in UTC (Coordinated Universal Time), when an operation occurred on an attribute of an object Resource.

2.2.1.8 Cause

A Cause is an Object Identifier of a Person, Request, or Workflow Instance that causes an attribute to be added to or deleted from a Resource.

2.2.1.9 Request Operation

A Request Operation is a Unicode string that represents the type of the request. The valid values that are used in this protocol are listed in the following table.

Value	Meaning
Create	Create new object Resource.
Put	Update an object Resource.
Get	Retrieve an object Resource.
Delete	Delete an object Resource.

2.2.1.10 Request Status

A Request Status is a Unicode string that represents the processing status of a request. The values that are used in this protocol are listed in the table in section 2.2.1.11.

2.2.1.11 Request Status Key

A Request Status key is a smallint that represents the processing status of a request. The values used in this protocol are listed in the following table.

Request Status	Request Status Key	Meaning
Canceled	9	The request has been canceled.

Request Status	Request Status Key	Meaning
Completed	14	The request completed successfully.
Committed	10	The request has been committed to the database.
Denied	0	The request failed authentication.
PostProcessing	12	The request has a workflow to run.
Validating	1	The request is being validated.
Validated	3	The request has been validated

2.2.1.12 Request Object Identifier

The Request Object Identifier is the Object Identifier of the object Resource that represents a request.

2.2.1.13 Rule Action Type

The Rule Action Type is a string that represents an action on an attribute that a Management Policy Rule gives permission to the principals to perform. The valid values that are used in this protocol are listed in the following table.

Value	Meaning
Read	Permission to read the attribute.
Create	Permission to create the attribute.
Modify	Permission to modify the attribute.
Delete	Permission to delete the attribute.
Add	Permission to Add value to a multivalued attribute.
Remove	Permission to Remove a value from a multivalued attribute.

2.2.1.14 Target Object Identifier

The Target Object Identifier is the Object Identifier of the Resource object the request is acting upon.

2.2.1.15 Workflow Instance Status

A Workflow Instance Status is a Unicode string that represents the status of a Workflow Instance. The values that are used in this protocol are listed in the table in section 2.2.1.16.

2.2.1.16 Workflow Instance Status Key

The Workflow Instance Status Key is a smallint that represents the current status of the Workflow instance. The valid values that are used in this protocol are listed in the following table.

Workflow Instance Status Key	Workflow Instance Status	Meaning
0	Running	Workflow Instance is Running
1	Completed	Workflow Instance is Completed
4	Created	Workflow Instance is Created
102	Pending	Workflow Instance is Pending

2.2.1.17 Workflow Control Message

The Workflow Control Message is a smallint that represents a message to the client that instructs it to do an action for that workflow. The valid values that are used in this protocol are listed in the following table.

Value	Meaning
0	None
1	Resume
2	Cancel

2.2.1.18 Request Display Name Template

The Request Display Name Template is a Unicode string that is used to create the Display Name attribute of the Request. The allowed values are listed in the following table.

Template	Usage
Create {0}: "{1}" Request	This template is used to create the Display Name of a create request.
Delete {0}: "{1}" Request	This template is used to create the Display Name of a delete request.
Update {0}: "{1}" Request	This template is used to create the Display Name of an update request.
Read {0}: "{1}" Request	This template is used to create the Display Name of a read request.

2.2.1.19 Host Identifier

The host identifier is a smallint value that uniquely identifies the client. It is allocated by calling the RegisterService stored procedure.

2.2.1.20 Workflow XOML

Workflow XOML is a constant string that defines a workflow. The value is:

```
<ns0:SequentialWorkflow x:Name="DomainSynchronizationWorkflow"
xmlns:ns0="clr-
namespace:Microsoft.ResourceManagement.Workflow.Activities;Assembly=Microsoft.ResourceManagem
ent,
           Version=4.0.2450.5, Culture=neutral, PublicKeyToken=31bf3856ad364e35"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/workflow">
```

```
  <ns0:ResolveGrammarActivity ResolvedExpression="{x:Null}"
GrammarExpression="[//Target/Domain]"
  x:Name="resolveGrammarActivity1" WorkflowDictionaryKey="TargetDomainAttribute" />
  <ns0:DomainSynchronizationActivity
  Domain="{ActivityBind resolveGrammarActivity1,Path=ResolvedExpression}"
  x:Name="domainSynchronizationActivity1" />
</ns0:SequentialWorkflow>
```

2.2.2 Bit Fields and Flag Structures

No common bit field or flag structures are defined in this protocol.

2.2.3 Binary Structures

No common binary structures are defined in this protocol.

2.2.4 Result Sets

2.2.4.1 DequeueWorkItem.ResultSet0

The DequeueWorkItem.ResultSet0 contains the future scheduled work items.

```
WorkItemId uniqueidentifier,
WorkItemData varbinary(max)
```

WorkItemId: A unique identifier for the work item. This MUST be ignored by the client.

WorkItemData: A binary that represents the work item. This MUST be ignored by the client.

2.2.4.2 GetObjectCurrent.ResultSet0

The GetObjectCurrent.ResultSet0 contains the Object Type Name of an object Resource.

```
Name nvarchar(448),
```

Name: The Object Type Name of an object Resource.

2.2.4.3 GetObjectCurrent.ResultSet1

The GetObjectCurrent.ResultSet1 contains the attribute data for attributes of an object Resource.

```
ObjectID uniqueidentifier,
AttributeKey smallint,
ValueBinary image,
ValueBoolean bit,
ValueDateTime datetime,
ValueInteger bigint,
ValueReference uniqueidentifier,
ValueString nvarchar(448),
ValueText ntext,
LocaleKey smallint,
AddTime datetime,
AddCause uniqueidentifier,
DeleteTime datetime,
DeleteCause uniqueidentifier,
```

ObjectID: The Object Identifier of the object Resource with which the attribute is associated.

AttributeKey: The Attribute Type Key of the attribute.

ValueBinary: The value of the attribute if the Attribute Data Type is Binary; otherwise, it is NULL.

ValueBoolean: The value of the attribute if the Attribute Data Type is Boolean; otherwise, it is NULL.

ValueDateTime: The value of the attribute if the Attribute Data Type is Datetime; otherwise, it is NULL.

ValueInteger: The value of the attribute if the Attribute Data Type is Integer; otherwise, it is NULL.

ValueReference: The value of the attribute if the Attribute Data Type is Reference; otherwise, it is NULL.

ValueString: The value of the attribute if the Attribute Data Type is String; otherwise, it is NULL.

ValueText: The value of the attribute if the Attribute Data Type is Text; otherwise, it is NULL.

LocaleKey: This MUST be NULL.

AddTime: The Time that the attribute was created.

AddCause: The Cause that created the attribute.

DeleteTime: The Time that the attribute was deleted, if the attribute has been deleted; otherwise, it is "9999-01-01 00:00:00.000".

DeleteCause: The Cause that deleted the attribute, if the attribute has been deleted; otherwise, it is NULL.

2.2.4.4 GetObjectCurrentRelative.ResultSet0

The GetObjectCurrentRelative.ResultSet0 contains the Object Type Name of an object Resource.

```
Name nvarchar(448),
```

Name: The Object Type Name of an object Resource.

2.2.4.5 GetObjectCurrentRelative.ResultSet1

The GetObjectCurrentRelative.ResultSet1 contains the attribute data for the attributes of an object Resource.

```
ObjectID uniqueidentifier,
AttributeKey smallint,
ValueBinary image,
ValueBoolean bit,
ValueDateTime datetime,
ValueInteger bigint,
ValueReference uniqueidentifier,
ValueString nvarchar(448),
ValueText ntext,
LocaleKey smallint,
AddTime datetime,
AddCause uniqueidentifier,
DeleteTime datetime,
DeleteCause uniqueidentifier,
```

ObjectID: The Object Identifier of the object Resource with which the attribute is associated.

AttributeKey: The Attribute Type Key of the attribute.

ValueBinary: The value of the attribute if the Attribute Data Type is Binary; otherwise, it is NULL.

ValueBoolean: The value of the attribute if the Attribute Data Type is Boolean; otherwise, it is NULL.

ValueDateTime: The value of the attribute if the Attribute Data Type is Datetime; otherwise, it is NULL.

ValueInteger: The value of the attribute if the Attribute Data Type is Integer; otherwise, it is NULL.

ValueReference: The value of the attribute if the Attribute Data Type is Reference; otherwise, it is NULL.

ValueString: The value of the attribute if the Attribute Data Type is String; otherwise, it is NULL.

ValueText: The value of the attribute if the Attribute Data Type is Text; otherwise, it is NULL.

LocaleKey: This MUST be NULL.

AddTime: The Time that the attribute was created.

AddCause: The Cause that created the attribute.

DeleteTime: The Time that the attribute was deleted, if the attribute has been deleted; otherwise, it is "9999-01-01 00:00:00.000".

DeleteCause: The Cause that deleted the attribute, if the attribute has been deleted; otherwise, it is NULL.

2.2.4.6 EvaluateNewRequest.ResultSet0

The EvaluateNewRequest.ResultSet0 contains information about Management Policy Rules and workflows that are required to be executed as a result of the request.

```
RuleIdentifier uniqueidentifier,
AuthenticationWorkflowDefinitionIdentifier uniqueidentifier,
AuthorizationWorkflowInstanceIdentifier uniqueidentifier,
ActionWorkflowInstanceIdentifier uniqueidentifier
```

RuleIdentifier: The Object Identifier of a Management Policy Rule that defines the associated workflows.

AuthenticationWorkflowDefinitionIdentifier: The Object Identifier of the Authentication Workflow Definition that is specified by the Management Policy Rule. It is NULL if the Management Policy Rule does not specify an Authentication Workflow.

AuthorizationWorkflowInstanceIdentifier: The Object Identifier of the Authorization Workflow instance that is created as a result of request. It is NULL if the Management Policy Rule does not specify an Authorization Workflow Definition.

ActionWorkflowInstanceIdentifier: The Object Identifier of the Workflow Instance that is created as a result of the request. It is NULL if the Management Policy Rule does not specify a Workflow Definition.

2.2.4.7 EvaluateNewRequest.ResultSet1

The EvaluateNewRequest.ResultSet1 contains information about the Set transitions that will occur as a result of executing the request.

```
ResourceIdentifier uniqueidentifier,
SetIdentifier uniqueidentifier,
Join bit,
Leave bit,
ActionWorkflowDefinitionIdentifier uniqueidentifier
```

ResourceIdentifier: The Object Identifier of the Resource object.

SetIdentifier: The Object Identifier of the Set that the object Resource is transitioning into or out of.

Join: Indicates whether the Resource is joining the set. It has a value of 1 if the Resource is joining the Set; otherwise, it is 0. If it is set to 1, then Leave MUST be 0. If it is set to 0, then Leave MUST be 1.

Leave: Indicates whether the Resource is leaving the set. It has a value of 1 if the Resource is leaving the Set; otherwise, it is 0.

ActionWorkflowDefinitionIdentifier: The Object Identifier of the Workflow Definition to run as a consequent of the object Resource that is transitioning into or out of the Set.

2.2.4.8 EvaluateRequest.ResultSet0

The EvaluateRequest.ResultSet0 contains the Object identifiers of the Workflow Definitions and Instances and the Object identifier of the associated Management Policy Rule that triggered these workflows as a result of executing the request.

```
RuleIdentifier uniqueidentifier,
AuthenticationWorkflowDefinitionIdentifier uniqueidentifier,
AuthorizationWorkflowInstanceIdentifier uniqueidentifier,
ActionWorkflowInstanceIdentifier uniqueidentifier
```

RuleIdentifier: The Object Identifier of the Management Policy Rule.

AuthenticationWorkflowDefinitionIdentifier: This MUST be NULL. This MUST be ignored by the client.

AuthorizationWorkflowInstanceIdentifier: This MUST be NULL. This MUST be ignored by the client.

ActionWorkflowInstanceIdentifier: The Object Identifier of the Workflow Instance.

2.2.4.9 EvaluateRequest.ResultSet1

The EvaluateRequest.ResultSet1 contains information about the Set transitions that will occur as a result of executing the request.

```
ResourceIdentifier uniqueidentifier,
SetIdentifier uniqueidentifier,
Join bit,
Leave bit,
ActionWorkflowDefinitionIdentifier uniqueidentifier
```

ResourceIdentifier: The Object Identifier of the Resource object.

SetIdentifier: The Object Identifier of the Set that the object Resource is transitioning into or out of.

Join: Indicates whether the Resource is joining the set. Has a value of 1 if the Resource is joining the Set; otherwise, it is 0. If it is set to 1, then Leave MUST be 0. If it is set to 0, then Leave MUST be 1.

Leave: Indicates whether the Resource is leaving the set. It has as value of 1 if the Resource is leaving the Set; otherwise, it is 0.

ActionWorkflowDefinitionIdentifier: The Object Identifier of the Workflow Definition, if any. Otherwise, it is NULL.

2.2.4.10 RetrieveExpiredTimerIds.ResultSet0

The RetrieveExpiredTimerIds.ResultSet0 contains workflow instance identifiers for workflow instances that have expired timers.

```
uidInstanceID uniqueidentifier,
```

uidInstanceID: Workflow instance identifier.

2.2.4.11 RetrieveWorkflowDataForHostActivator.ResultSet0

The RetrieveWorkflowDataForHostActivator.ResultSet0 contains Workflow Definition information.

```
WorkflowDefinitionIdentifier uniqueidentifier,
WorkflowDefinitionVersionKey int,
WorkflowDefinitionXOML nvarchar(max),
WorkflowDefinitionRules nvarchar(max),
WorkflowDefinitionDeleteCause uniqueidentifier,
ActiveHostedWorflowDefinitionSequenceNumber int
```

WorkflowDefinitionIdentifier: The Object Identifier of the Workflow Definition.

WorkflowDefinitionVersionKey: Specifies the version of the Workflow Definition.

WorkflowDefinitionXOML: This MUST be Workflow XOML.

WorkflowDefinitionRules: This MUST be NULL.

WorkflowDefinitionDeleteCause: The Cause that deleted the Workflow Definition. It is NULL if the workflow Definition has not been deleted.

ActiveHostedWorkflowDefinitionSequenceNumber: The sequence number of the Workflow Definition in an enumeration of Workflow Definitions.

2.2.4.12 RetrieveWorkflowDataForHostActivator.ResultSet1

The RetrieveWorkflowDataForHostActivator.ResultSet1 contains the list of the workflow instances that belong to this host and whose status is not Completed, Terminated, or Canceled.

```
WorkflowInstanceIdentifier uniqueidentifier,
WorkflowInstanceDefinitionKey int,
WorkflowInstanceStatusKey smallint,
WorkflowInstanceHostControlMessageKey tinyint,
```

WorkflowInstanceIdentifier: The Object Identifier of the Workflow Instance.

WorkflowInstanceDefinitionKey: The Workflow Definition to which this instance relates.

WorkflowInstanceStatusKey: The key of the Workflow Instance Status.

WorkflowInstanceHostControlMessageKey: The key of the Workflow Control Message.

2.2.4.13 RetrieveWorkflowDataForHostActivator.ResultSet2

The RetrieveWorkflowDataForHostActivator.ResultSet2 contains identifiers for the Requests that required re-dispatch.

```
requestIdentifier uniqueidentifier,
```

requestIdentifier: The Request Object Identifier.

2.2.4.14 RetrieveWorkflowInstance.ResultSet0

The RetrieveWorkflowInstance.ResultSet0 contains Workflow Instance information.

```
Identifier uniqueidentifier,
DefinitionIdentifier uniqueidentifier,
DefinitionKey int,
WorkflowType smallint,
RequestIdentifier uniqueidentifier,
RequestorIdentifier uniqueidentifier,
TargetIdentifier uniqueidentifier,
Status smallint,
```

Identifier: The Object Identifier of the Workflow Instance Object Resource.

DefinitionIdentifier: The Object Identifier of the Workflow Definition.

DefinitionKey: A unique value that identifies the Workflow Definition.

WorkflowType: This MUST be 30.

RequestIdentifier: The Request Object Identifier of the Request that caused the Workflow Instance to be created.

RequestorIdentifier: The Cause of the Request.

TargetIdentifier: The Target Object Identifier.

Status: The Workflow Instance Status.

2.2.4.15 UpdateRequest.ResultSet0

The UpdateRequest.ResultSet0 contains information about whether a request requires reevaluation and whether it has any associated child request.

```
ReEvaluation bit,
RequestDependencyKey  tinyint
```

ReEvaluation: Has a value of 1 if the request requires reevaluation; otherwise, it is 0.

RequestDependencyKey: Has a value of 1 if the request has one or more child requests; otherwise, it is 0.

2.2.4.16 UpdateRequest.ResultSet1

The UpdateRequest.ResultSet1 contains Management Policy Rules against which the request is required to be reevaluated.

```
PolicyRuleIdentifier uniqueidentifier
```

PolicyRuleIdentifier: The Object Identifier of the Management Policy Rule.

2.2.4.17 UpdateRequest.ResultSet2

The UpdateRequest.ResultSet2 contains Object Identifiers of Workflow Instances that are required to be run during processing of a request.

```
WorkflowInstanceIdentifier
 uniqueidentifier
```

WorkflowInstanceIdentifier: The Object Identifier of the Workflow Instance.

2.2.4.18 UpdateRequest.ResultSet3

The UpdateRequest.ResultSet3 contains information about Set transitions that will occur as a result of processing the update request.

```
ResourceIdentifier uniqueidentifier,
SetIdentifier  uniqueidentifier,
Join bit,
Leave bit,
ActionWorkflowDefinitionIdentifier uniqueidentifier
```

ResourceIdentifier: The Object Identifier of a Resource.

SetIdentifier: The Object Identifier of the Set that the object Resource is transitioning into or out of.

Join: Indicates whether the Resource is joining the Set. The value is 1 if the Resource is joining the Set; otherwise, it is 0. If it is set to 1, then Leave MUST be 0. If it is set to 0, then Leave MUST be 1.

Leave: Indicates whether the Resource is leaving the Set. The value is 1 if the Resource is leaving the Set; otherwise, it is 0.

ActionWorkflowDefinitionIdentifier: The Object Identifier of the Workflow Definition.

2.2.4.19 UpdateRequestStatusForCompletedActionWorkflows.ResultSet0

The UpdateRequestStatusForCompletedActionWorkflows.ResultSet0 contains Object Identifiers of requests that have been processed by this stored procedure, along with their statuses.

```
RequestIdentifier uniqueidentifier,
RequestStatus nvarchar(448),
```

RequestIdentifier: Request Object Identifier.

RequestStatus: Request Status.

2.2.4.20 UpdateRequestStatusForCompletedActionWorkflows.ResultSet1

The UpdateRequestStatusForCompletedActionWorkflows.ResultSet1 contains Object Identifiers of child requests that were processed by this stored procedure, along with their statuses.

```
RequestIdentifier uniqueidentifier,
RequestStatus nvarchar(448),
```

RequestIdentifier: Request Object Identifier.

RequestStatus: Request Status.

2.2.4.21 UpdateRequestStatusForCompletedActionWorkflows.ResultSet2

The UpdateRequestStatusForCompletedActionWorkflows.ResultSet2 contains Object Identifiers of requests that failed to be updated during execution of the stored procedure.

```
RequestIdentifier uniqueidentifier,
```

RequestIdentifier: Request Object Identifier.

2.2.5 Tables and Views

2.2.5.1 ObjectsInternal

The ObjectsInternal table contains every attribute for all object Resources.

```
ObjectID uniqueidentifier NOT NULL,
ObjectTypeKey smallint NOT NULL,
AttributeKey smallint NOT NULL,
ValueBinary varbinary(max) NULL,
ValueBoolean bit NULL,
ValueDateTime datetime NULL,
ValueInteger bigint NULL,
ValueReference uniqueidentifier NULL,
ValueString nvarchar(448) NULL,
ValueText nvarchar(max) NULL,
AddTime datetime NOT NULL,
AddCause uniqueidentifier NOT NULL,
DeleteTime datetime NOT NULL,
DeleteCause uniqueidentifier NULL,
LocaleKey smallint NULL,
ExportEmbargo smallint NULL,
ObjectKey bigint IDENTITY(1,1) NOT NULL
```

ObjectID: The Object Identifier of an object Resource.

ObjectTypeKey: The Object Type Key.

AttributeKey: The Attribute Type Key.

ValueBinary: The value of the attribute if the Attribute Data Type is Binary; otherwise, it is NULL.

ValueBoolean: The value of the attribute if the Attribute Data Type is Boolean; otherwise, it is NULL.

ValueDateTime: The value of the attribute if the Attribute Data Type is Datetime; otherwise, it is NULL.

ValueInteger: The value of the attribute if the Attribute Data Type is Integer; otherwise, it is NULL.

ValueReference: The value of the attribute if the Attribute Data Type is Reference; Otherwise, it is NULL. This value is the Object Identifier of another object Resource.

ValueString: The value of the attribute if the Attribute Data Type is String; otherwise, it is NULL.

ValueText: The value of the **attribute** if the Attribute Data Type is Text; otherwise, it is NULL.

AddTime: The Time when the value was added.

AddCause: The Cause that added the value.

DeleteTime: The Time when the value was deleted. If the attribute has not been deleted, the value is "9999-01-01 00:00:00.000".

DeleteCause: The Cause that deleted this value. If the attribute has not been deleted, the value is NULL.

LocaleKey: This MUST be ignored by the client.

ExportEmbargo: This MUST be ignored by the client.

ObjectKey: A key that is unique within the scope of this table that represents a row of this table. This MUST be ignored by the client.

2.2.5.2 BindingInternal

The BindingInternal table contains the mapping that specifies the set of attribute types that are allowed for each object type.

```
ObjectTypeKey smallint NOT NULL,
AttributeKey smallint NOT NULL,
Required bit NOT NULL,
BdObjectID uniqueidentifier NOT NULL,
ObjectType nvarchar(448) NOT NULL,
AttributeName nvarchar(448) NOT NULL,
StringRegex nvarchar(max) NULL,
IntegerMinimum bigint NULL,
IntegerMaximum bigint NULL,
```

ObjectTypeKey: The Object Type Key.

AttributeKey: The Attribute Type Key.

Required: Specifies whether the attribute identified by AttributeKey is required or optional for objects whose object type is identified by ObjectTypeKey. If the value is 1, the attribute is required. If it is 0, the attribute is optional.

BdObjectID: The Object Identifier for this Binding.

ObjectType: The Object Type Name.

AttributeName: The Attribute Type Name. The AttributeName and AttributeKey MUST be for the same Attribute Type Description.

StringRegex: A Regular Expression for the valid values of the attribute. The value is NULL if no validation is required. For more information about Regular Expression, see [MSDN-RegEx].

IntegerMinimum: The minimum permitted value of the attribute if the Attribute Data Type of the attribute identified by AttributeKey is Integer. The value is NULL if there is no minimum or if the Attribute Data Type of the attribute is not Integer.

IntegerMaximum: The maximum permitted value of the attribute if the Attribute Data Type of the attribute identified by AttributeKey is Integer. The value is NULL if there is no maximum or if the Attribute Data Type of the attribute is not Integer.

2.2.5.3 ManagementPolicyEnforcementRead

The ManagementPolicyEnforcementRead is a view that contains the read access permissions for the attributes in the ObjectsInternal table.

```
ObjectID uniqueidentifier NOT NULL,
Principal uniqueidentifier NULL,
Resource uniqueidentifier NULL,
ActionParameterKey smallint NULL,
```

ObjectID: The Object Identifier of Management Policy Rule that gives this permission.

Principal: The Object Identifier of the Set that contains the Persons that have read access permission.

Resource**:** The Target Object Identifier of the Resource to which the members of the Principal Set have read access.

ActionParameterKey: The Attribute Type Key of an attribute of the Resource object that is identified by Resource to which the members of the Principal Set have read access.

2.2.5.4 Attributes

The Attributes view contains information about the Attribute Type Descriptions.

```
Key smallint NOT NULL,
Name nvarchar(448) NOT NULL,
AtdObjectID uniqueidentifier NOT NULL,
DataType nvarchar(128) NOT NULL,
Multivalued bit NOT NULL,
DisplayName nvarchar(448) NULL,
Description nvarchar(448) NULL,
```

Key: The Attribute Type Key.

Name: The Attribute Type Name.

AtdObjectID: The Object Identifier of the Attribute Type Description.

DataType: The Attribute Data Type.

Multivalued: If the attribute is Multivalued, this attribute MUST be 1; otherwise, this attribute MUST be 0.

DisplayName: The Display Name of Attribute Type Description.

Description: The Description of Attribute Type Description.

2.2.5.5 SchemaObject

The SchemaObject view contains information about the Object Type Descriptions.

```
ObjectTypeKey smallint NOT NULL,
Name nvarchar(448) NOT NULL,
DisplayName nvarchar(448) NULL,
```

ObjectTypeKey: The Object Type Key.

Name: The Object Type Name.

DisplayName: The Display Name of the Object Type Description.

2.2.6 XML Structures

2.2.6.1 Namespaces

This specification defines and references various [XML namespaces] using the mechanisms specified in [XMLNS-2ED]. Although this specification associates a specific [XML namespace] prefix for each [XML namespace] that is used, the choice of any particular [XML namespace] prefix is implementation-specific and not significant for interoperability.

Prefix	Namespace URI	Reference
xs	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] and [XMLSCHEMA2]
xsd	http://www.w3.org/2001/XMLSchema	[XMLSCHEMA1] and [XMLSCHEMA2]
xsi	http://www.w3.org/2001/XMLSchema-instance	[XMLSCHEMA1]

2.2.6.2 Simple Types

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

2.2.6.3 Complex Types

None.

2.2.6.3.1 CreateRequestParameter

CreateRequestParameter is an XML Type that represents a Request Parameter as part of create resource attribute.

```
<xs:complexType name="CreateRequestParameter">
      <xs:sequence>
        <xs:element name="Calculated" type="xs:boolean" />
        <xs:element name="PropertyName" type="xs:string" />
        <xs:element name="Value" type="xs:string" />
        <xs:element minOccurs="0" name="Mode" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
```

Calculated: This MUST be false.

PropertyName: The Attribute Type Name of the attribute to be created or updated.

Value: The value to set for the attribute.

Mode: Specifies the operation to perform on the attribute. The allowed values for this parameter are listed in the following table.

Value	Meaning
Insert	Adds a value to Multivalued attribute.

Value	Meaning
Remove	Deletes a value from a Multivalued attribute.
Modify	Modifies the value of an attribute that is not Multivalued.

2.2.6.3.2 UpdateRequestParameter

UpdateRequestParameter is an XML Type that represents a Request Parameter to update resource.

```
 <xs:complexType name="UpdateRequestParameter">
      <xs:sequence>
        <xs:element name="Calculated" type="xs:boolean" />
        <xs:element name="PropertyName" type="xs:string" />
        <xs:element name="Value" type="xs:string" />
        <xs:element minOccurs="0" name="Mode" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
```

Calculated: This MUST be false.

PropertyName: The Attribute Type Name of the attribute to be created or updated.

Value: The value to set for the attribute.

Mode: Specifies the operation to perform on the attribute. The allowed values for this parameter are listed in the following table.

Value	Meaning
Insert	Adds a value to Multivalued attribute.
Remove	Deletes a value from a Multivalued attribute.
Modify	Modifies the value of an attribute that is not Multivalued.

2.2.6.4 Elements

The following table summarizes the set of common XML Schema element definitions defined by this specification. XML Schema element definitions that are specific to a particular operation are described with the operation.

Element	Description
RequestParameters	The RequestParameters descripes one or more Request parameter which each describes the requested action on an attribute.
ValueList	The ValueList structure represents a list of **strings**.
Membership Condition	The MembershipCondition element contains the identifiers of the Membership Condition Statements that constitute the membership condition.
Membership Condition Statement	The Membership Condition Statement element, defined by the Statement element, contains the attributes of the Membership Condition Statement

Element	Description
	Statements from which the predicate constitutes.
Set Transition	The SetTransition element contains the Object Identifier of a Resource that underwent a Set Transition during processing of a Request.
WorkflowInstances	The WorkflowInstances element contains the ObjectIdentifiers of WorkflowInstance Resources.

2.2.6.4.1 RequestParameters

RequestParameters is an XML schema element that describes the request parameters. Each Request has several request parameters. Each parameter describes the requested action on an attribute.

The RequestParameters for a create request:

```
<xsd:element name="RequestParameters">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="RequestParameter" minOccurs="0" maxOccurs="1"
type="xsi:CreateRequestParameter" />
       </xsd:sequence>
   </xsd:complexType>
</xsd:element>
The RequestParameters for an update request:
<xsd:element name="RequestParameters">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="RequestParameter" minOccurs="0" maxOccurs="1"
type="xsi:UpdateRequestParameter" />
       </xsd:sequence>
</xsd:complexType>
</xsd:element>
```

Example:

```
<RequestParameters>
  <RequestParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="CreateRequestParameter">
    <Calculated>false</Calculated>
    <PropertyName>ObjectType</PropertyName>
    <Value xsi:type="xsd:string">Person</Value>
  </RequestParameter>
  <RequestParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="CreateRequestParameter">
    <Calculated>false</Calculated>
    <PropertyName>DisplayName</PropertyName>
    <Value xsi:type="xsd:string">redmond\pkmuser2</Value>
  </RequestParameter>
  <RequestParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="CreateRequestParameter">
    <Calculated>false</Calculated>
    <PropertyName>AccountName</PropertyName>
    <Value xsi:type="xsd:string">pkmuser2</Value>
  </RequestParameter>
  <RequestParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="CreateRequestParameter">
    <Calculated>false</Calculated>
    <PropertyName>Domain</PropertyName>
    <Value xsi:type="xsd:string">redmond</Value>
  </RequestParameter>
  <RequestParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="CreateRequestParameter">
    <Calculated>false</Calculated>
    <PropertyName>FirstName</PropertyName>
    <Value xsi:type="xsd:string">redmond\pkmuser2</Value>
  </RequestParameter>
  <RequestParameter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xsi:type="CreateRequestParameter">
    <Calculated>false</Calculated>
    <PropertyName>MailNickname</PropertyName>
    <Value xsi:type="xsd:string">pkmuser2</Value>
  </RequestParameter>
</RequestParameters>
```

2.2.6.4.2 ValueList

ValueList is a structure that represents a list of strings.

```
<xs:element name="v" type="xs:string" minOccurs="1" maxOccurs="unbound"/>
```

Example:

```
<v>ObjectType</v><v>DisplayName</v><v>Description</v>
```

2.2.6.4.3 Membership Condition

The MembershipCondition element contains the identifiers of the Membership Condition Statements that constitute the membership condition.

```
  <xs:element name="MembershipCondition">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Statements">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="Identifier" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

MembershipCondition.Statements: A Sequence of Object Identifiers of Membership Condition Statement.

```
Example:
<MembershipCondition>
  <Statements>
    <Identifier>a2cde1a9-9168-4dff-bbc5-4f4227b530d3</Identifier>
  </Statements>
</MembershipCondition>
```

2.2.6.4.4 Membership Condition Statement Attribute

The Membership Condition Statement element, defined by the Statement element, contains the attributes of the Membership Condition Statements from which the predicate constitutes.

```
<xs:element name="Statement">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Attributes">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Attribute" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
</xs:element>
```

The value that is used in this protocol MUST be:

```
<Statement>
  <Attributes>
    <Attribute>ObjectType</Attribute>
  </Attributes>
</Statement>
```

2.2.6.4.5 Set Transition

The SetTransition element contains the Object Identifier of a Resource that underwent a Set Transition during processing of a Request.

```
<xs:element name="SetTransition">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ResourceIdentifier" type="xs:string" />
        <xs:element name="SetIdentifier" type="xs:string" />
        <xs:element name="Join" type="xs:boolean" />
        <xs:element name="Leave" type="xs:boolean" />
        <xs:element name="ActionWorkflowDefinitionIdentifier" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

SetTransition.ResourceIdentifier: The Resource Object Identifier

SetTransition.SetIdentifier: The Object Identifier of the Set.

SetTransition.Join: True, if the Resource is joining the Set; otherwise, it is false.

SetTransition.Leave: True, if the Resource is leaving the Set; otherwise, it is false.

SetTransition.ActionWorkflowDefinitionIdentifier: The Object Identifier of the Workflow Definition that requires to be run as result of Joining or leaving the Set.

```
Example:

<SetTransition xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ResourceIdentifier>640F3FB1-7836-43CB-8A73-BF914986FE9E</ResourceIdentifier>
  <SetIdentifier>47883536-CEF0-4759-BD58-296788A4548C</SetIdentifier>
  <Join>true</Join>
  <Leave>false</Leave>
  <ActionWorkflowDefinitionIdentifier>955E3366-FBCC-43EE-B6E4-
2001B81971DA</ActionWorkflowDefinitionIdentifier>
</SetTransition>
```

2.2.6.4.6 WorkflowInstances

The WorkflowInstances element contains the ObjectIdentifiers of WorkflowInstance Resources.

```
<xs:element name="WorkflowInstances">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Identifier" minOccurs="0" maxOccurs="unbounded" type="xs:string" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
Example:
<WorkflowInstances>
  <Identifier>8354bc44-4ebc-469c-9b5f-ec3608295d26</Identifier>
</WorkflowInstances>
```

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.

3 Protocol Details

3.1 Server Details

3.1.1 Abstract Data Model

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

3.1.1.1 Logical Data Model

The following Unified Modeling Language (UML) diagram illustrates the Logical data model.

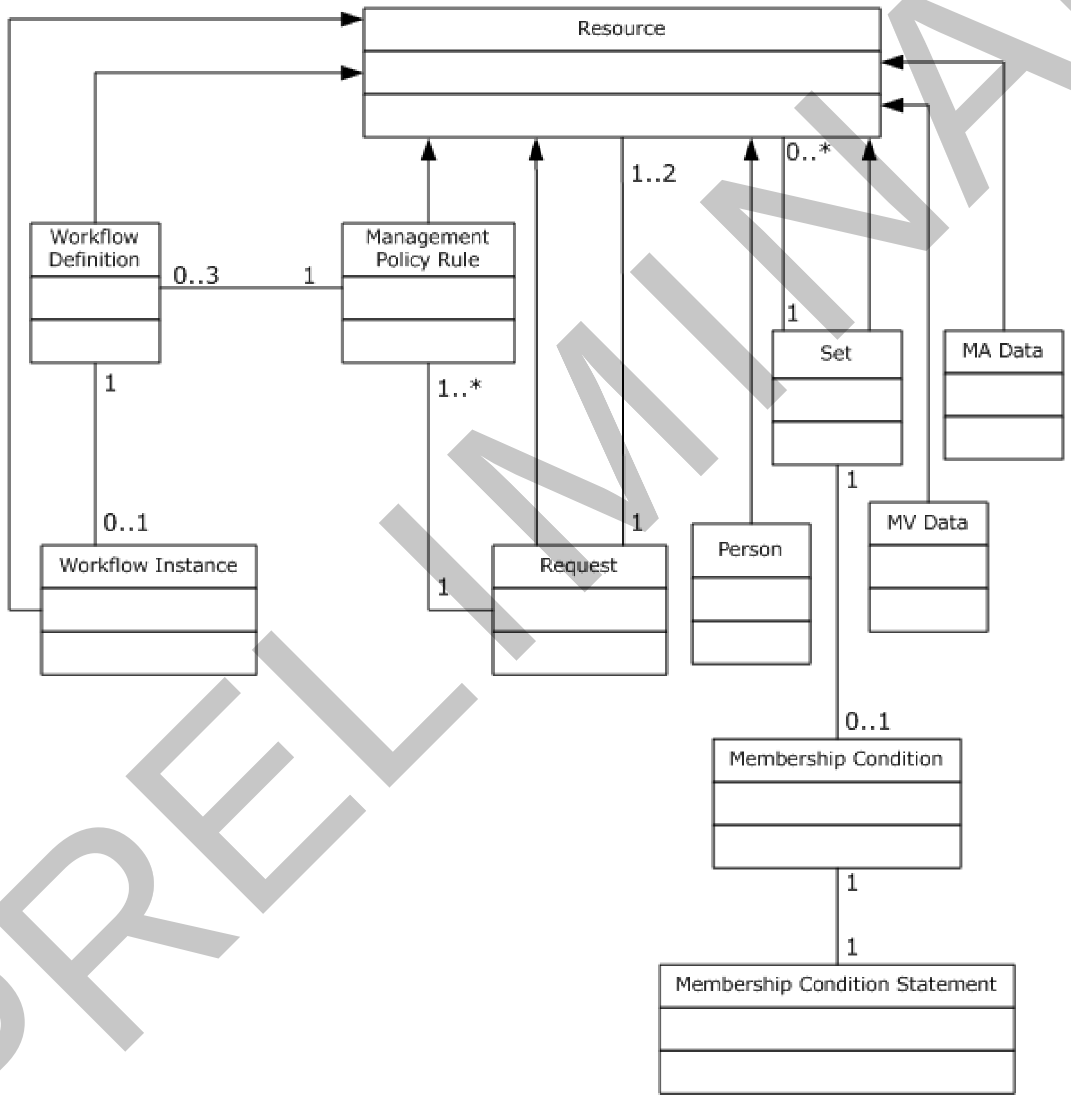

Figure 3: Logical data model

3.1.1.1.1 Resource

Objects stored in the Configuration Database are called Resources. Each object Resource consists of several attributes, and each object is represented by a row in the ObjectsInternal table. The protocol supports the list of Resources that are identified by Object Type Key, as described in section 2.2.1.2. Each Resource has a common set of attributes, as defined in section 2.2.1.6.1.

3.1.1.1.2 Person

A Person is a Resource that holds the information associated with a user. The attributes associated with a Person are defined in section 2.2.1.6.8.

3.1.1.1.3 Request

A Request is a Resource that represents a request to perform an operation on one or more attributes of a Resource. The Resource that the Request directly operates against is called the **Target Object Resource**. The processing of a Request can lead to the creation of one or more other Requests, which are called child Requests. The attributes associated with a Request are defined in section 2.2.1.6.9.

3.1.1.1.4 Set

A Set is a named collection of Resources that are identified by a given filter expression that is associated with the Set or an explicit list of object Resources. The attributes associated with a Set are defined in section 2.2.1.6.10.

The server MUST contain the following Sets that have the following attribute values.

Sync Configuration Administrators Set

Attribute	Value
ObjectType	Set
Description	Sync Configuration Administrators
DisplayName	Sync Configuration Administrators

Management Agent Data Objects Set

Attribute	Value
ObjectType	Set
Description	Management Agent Data Objects
DisplayName	Management Agent Data Objects
Filter	<?xml version="1.0" encoding="utf-16"?> <Filter xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" Dialect="http://schemas.microsoft.com/2006/11/XPathFilterDialect" xmlns="http://schemas.xmlsoap.org/ws/2004/09/enumeration">/ma-data</Filter>

Attribute	Value
Temporal	0

3.1.1.1.5 Membership Condition

A Membership Condition is a container for the identifiers of Membership Condition Statements that defines the predicate of the Filter of a Set. The fields associated with Membership Condition are listed on the following table.

Field	Description
SetIdentifier	A unique identifier for the Set
MemebershipConditionIdentifier	A unique identifier for the Membership Condition
MemebershipConditionStatement	An XML element that defines the Membership Condition Statement identifiers, as described in section 2.2.6.4.3

The server MUST contain the following Membership Conditions that have the following attribute values:

Attribute	Value
SetIdentifier	The valueReference of the ObjectId attribue of the set that has a DisplayName "Management Agent Data Objects"
MemebershipConditionStatement	An XML element that has the ManagementAgentDataObjectsStatement identifier that is defined on section 3.1.1.1.7

3.1.1.1.6 Membership Condition Statement

A Membership Condition Statement is an internal representation of a predicate in a Membership Condition. The fields associated with a Membership Condition Statement are listed in the following table.

Field	Description
StatementIdentifier	A unique identifier for the Membership Condition Statement.
Xpath	An XPath expression that defines the condition statement. This MUST be "ObjectType=ma-data" or "ObjectType=mv-data".
Attribute	The Attribute Type Name of an attribute. This MUST be "ObjectType".
OperatorEqual	A Boolean to indicate that the operator is equal. This MUST be 1.
literalValueString	The value that the result of the Xpath expression is compared against to determine equality. This MUST be "ma-data" or "mv-data".
Attributes	The Membership Condition Statement Attribute, as defined in section 2.2.6.4.4.

The server MUST contain the following Membership Condition Statement that has the following attribute values:

Attribute	Value
Xpath	ObjectType=ma-data
Attribute	ObjectType
OperatorEqual	1
literalValueString	ma-data

Attributes	`<Statement>`
	`  <Attributes>`
	`    <Attribute>ObjectType</Attribute>`
	`  </Attributes>`
	`</Statement>`

3.1.1.1.7 Management Policy Rule

A Management Policy Rules stipulates access policy for Resources. The attributes associated with a Management Policy Rule are defined in section 2.2.1.6.6.

A Management Policy Rule with a value of 1 in the ValueBoolean field of the GrantRight attribute permits the members of the PrincipalSet attribute to perform the actions defined by the ActionType attribute on attributes that are defined by the ActionParameter attribute on Resources that are members of the Set identified by the ResourceCurrentSet attribute of the Management Policy Rule.

A Management Policy Rule with a value of 0 in the ValueBoolean field of the GrantRight attribute and that has an ActionWorkflow attribute defines a workflow that is to be run during the Request Processing.

The server MUST contain a Management Policy Rule Resource that has the following attribute values.

Attribute	Value
ObjectType	ManagementPolicyRule
Description	All Sync Configuration Administrators can update Management Agent Data Objects
DisplayName	All Sync Configuration Administrators can update Management Agent Data Objects
ActionParameter	*
ActionType	Create
ActionType	Delete
ActionType	Modify
ActionType	Read
ActionType	Add
ActionType	Remove
GrantRight	1

Attribute	Value
Disabled	0
PrincipalSet	The valueReference of the ObjectId attribue of the set that has a DisplayName "Sync Configuration Administrators"
ResourceCurrentSet	The valueReference of the ObjectId attribue of the set that has a DisplayName "Management Agent Data Objects"
ResourceFinalSet	The valueReference of the ObjectId attribue of the set that has a DisplayName "Management Agent Data Objects"

3.1.1.1.8 Workflow Definition

A Workflow Definition is a Resource that holds information that is associated with the definition of a workflow. The attributes associated with a Workflow Definition are defined in section 2.2.1.6.11.

The server MUST maintain a table of Workflow Definitions where each entry consists of a unique sequence number and the Workflow XOML of the workflow.

The server MUST contain Workflow Definition Resource that has the following attributes values.

Attribute	Value
ObjectType	WorkflowDefinition
DisplayName	Domain / DomainConfiguration Synchronization Workflow
RequestPhase	Action
RunOnPolicyUpdate	0
XOML	Workflow XOML

This Workflow Definition has sequence number 1.

3.1.1.1.9 Workflow Instance

A Workflow Instance is a Resource that holds information that is associated with a workflow instance, as defined in section 2.2.1.6.12. When a Request matches a Management Policy Rule that has an ActionWorkflow, a Workflow Instance is created and its Object Identifier is returned to the client, as defined in sections EvaluateNewRequest and EvaluateRequest stored procedures.

3.1.1.1.10 MA-Data

An MA-DATA is a Resource that holds the information that is associated with a management agent data structure. The attributes associated with an MA-Data are defined in section 2.2.1.6.5.

3.1.1.1.11 MV-Data

An MV-DATA is a Resource that holds the information that is associated with a Metaverse Configuration Data. The attributes associated with an MV-Data are defined in section 2.2.1.6.7.

3.1.1.2 Schema

As illustrated in the previous section, each Resource consists of a collection of attributes. The collection of Resource types that are supported by the server, and the set of attributes associated with each Resource type, is defined through a set of schema Resources. The following figure illustrates the schema Resources.

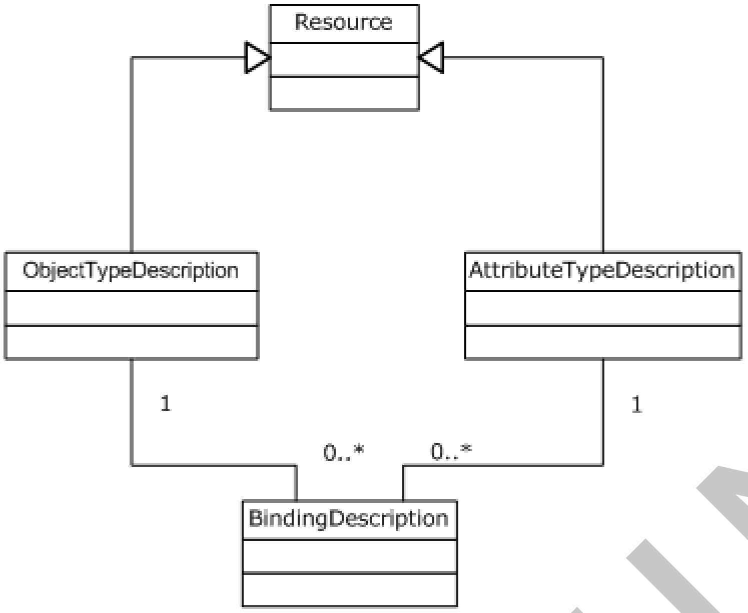

Figure 4: Schema Resources

3.1.1.2.1 ObjectTypeDescription

An ObjectTypeDescription is a Resource that holds the information that is related to a given Object Type. The set of Object Types that are supported by the protocol is defined in section 2.2.1.3. The attributes associated with an ObjectTypeDescription are defined in section 2.2.1.6.2.

3.1.1.2.2 AttributeTypeDescription

An AttributeTypeDescription is a Resource that holds the information that is related to a given attribute. The set of attributes that are supported by the protocol are defined in section 2.2.1.6. The attributes associated with an AttributeTypeDescription are defined in section 2.2.1.6.3.

3.1.1.2.3 BindingDescription

A BindingDescription is a Resource that holds the information that is related to the binding between an ObjectTypeDescription and an AttributeTypeDescription. A binding between an ObjectTypeDescription and an AttributeTypeDescription specifies that Resources whose ObjectType matches the ObjectTypeDescription Name have an attribute as defined by the AttributeTypeDescription. The attributes associated with a BindingTypeDescription are defined in section 2.2.1.6.4.

3.1.1.3 Mapping Logical Model to Abstract Data Model

The server MUST maintain the following tables:

- ObjectsInternal table
- MembershipCondition table
- MembershipConditionStatements table
- RegisteredService table

ObjectsInternal Table

A Resource consists of a set of attributes. Each attribute can be represented by a row in a table where a column specifies the ObjectID of the Resource to which the attribute belongs, and another column specifies the AttributeType of the attribute. In this way, all Resources can be stored in a single table. The schema for this table is the same, as defined in section 2.2.5.1, which provides a direct mapping between that table and the ADM.

MembershipCondition table

Entries in the Membership Condition table contain the fields that are defined in section 3.1.1.1.5.

MembershipConditionStatement table

Entries in the Membership Condition Statement table contain the fields that are defined in section 3.1.1.1.6.

RegisteredService table

Entries in the RegisteredService table contain the fields that are defined in the following table.

Field	Description
HostName	A string that represents the machine name of the protocol client.
HostId	An integer value that is assigned by the server to that client through a call to the RegisterServer stored procedure, as defined in section 3.1.5.20.

3.1.2 Timers

An execution timeout timer is set up on the Configuration Database to govern the execution time for any stored procedure call or select statement. The amount of time is governed by a SQL server connection time-out value configured on the Configuration Database for all connections.

3.1.3 Initialization

A connection to the SQL server that uses the underlying protocol layers that are specified in section 1.4 MUST be established before using this protocol, as specified in [MS-TDS].

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The T-SQL syntax for each Stored procedure and Result Set, and the variables of which they are composed, is defined in [MSTSQL]. In the T-SQL syntax, the variable name is followed by the type of the variable, which can optionally have a length value in brackets and can optionally have a default value indicated by an equal sign, followed by the default value. Unless otherwise specified, all Stored Procedures defined in this section are located in the Configuration Database.

For definitional clarity, a name has been assigned to any columns in the result sets that do not have a defined name in their current implementation. This does not affect the operation of the result set, because the User Profile Synchronization Configuration Service can access any column with no defined name by ordinal position. Such names are designated in the text by using braces in the form {name}. For interoperability, named columns in Result Sets are specified with what they are named, and columns marked with braces have no defined name. User Profile Synchronization Configuration Service implementations MUST NOT rely on any column name in a Result Set. All functions, result sets, and stored procedures are defined using T-SQL.

3.1.5.1 AddObjectValueBinary

The AddObjectValueBinary stored procedure is invoked to add an attribute whose Attribute Data Type is Binary, to a specified object Resource. This attribute is added as a row in the ObjectsInternals table.

```
PROCEDURE AddObjectValueBinary (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value varbinary(max),
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object specified by @ObjectID, and whose Attribute Data Type is Binary.

@Value: The value of the attribute being added. If the object identified by @ObjectID already has an attribute that is identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueBinary field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set the AddCause field of the attribute. This MUST NOT be NULL.

@Time: A Time value to be set for the AddTime field of the attribute. If the value is NULL, the server MUST set the AddTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.5.2 AddObjectValueBoolean

The AddObjectValueBoolean stored procedure is invoked to add an attribute whose Attribute Data Type is Boolean, to a specified object Resource. This attribute is added as a row in the ObjectsInternals table.

```
PROCEDURE AddObjectValueBoolean (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value bit,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object specified by @ObjectID, and whose Attribute Data Type is Boolean.

@Value: The value of the attribute. If the object identified by @ObjectID already has an attribute identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueBoolean field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set the AddCause field of the attribute. This MUST NOT be NULL.

@Time: A Time value to be set for the AddTime field of the attribute. If the value is NULL, the server MUST set the AddTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.3 AddObjectValueDateTime

The AddObjectValueDateTime stored procedure is invoked to add an attribute whose Attribute Data Type is DateTime, to a specified object Resource. This attribute is added as a row in the ObjectsInternals table.

```
PROCEDURE AddObjectValueDateTime (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value datetime,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object specified by @ObjectID, and whose Attribute Data Type is DateTime.

@Value: The value of the attribute. If the object identified by @ObjectID already has an attribute that is identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueDateTime field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set the AddCause field of the attribute. This MUST NOT be NULL.

@Time: A Time value to be set for the AddTime field of the attribute. If the value is NULL, the server MUST set the AddTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.5.4 AddObjectValueInteger

The AddObjectValueInteger stored procedure is invoked to add an attribute whose Attribute Data Type is Integer, to a specified object Resource. This attribute is added as a row in the ObjectsInternals table.

```
PROCEDURE AddObjectValueInteger (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value bigint,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object that is specified by @ObjectID, and whose Attribute Data Type is Integer.

@Value: The value of the attribute. If the object identified by @ObjectID already has an attribute that is identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueInteger field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set for the AddCause field of the attribute. This MUST NOT be NULL.

@Time: A Time value to be set for the AddTime field of the attribute. If NULL, the server MUST set the AddTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.5 AddObjectValueReference

The AddObjectValueInteger stored procedure is invoked to add an attribute whose Attribute Data Type is Reference, to a specified object Resource. This attribute is added as a row in the ObjectsInternals table.

```
PROCEDURE AddObjectValueReference (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value uniqueidentifier,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added to. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute that is being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object specified by @ObjectID, and whose Attribute Data Type is Reference.

@Value: The value of the attribute. If the object identified by @ObjectID already has an attribute that is identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueReference field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set for the AddCause field of the attribute. This MUST NOT be NULL.

@Time: A Time value to be set for the AddTime field of the attribute. If the value is NULL, the server MUST set the AddTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.6 AddObjectValueString

The AddObjectValueString stored procedure is invoked to add an attribute whose Attribute Data Type is String, to a specified object Resource. This attribute is added as a row in the ObjectsInternals table.

```
PROCEDURE AddObjectValueString (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value nvarchar(448),
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object that is specified by @ObjectID, and whose Attribute Data Type is String.

@Value: The value of the attribute. If the object that is identified by @ObjectID already has an attribute identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueString field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set for the AddCause field of the attribute. This MUST NOT be NULL.

@Time: The Time at which the new value is created. If the time value is NULL, the server will assign the current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.7 AddObjectValueText

The AddObjectValueText stored procedure is invoked to add an attribute whose Attribute Data Type is Text, to a specified object Resource. This attribute is added as a row in to the ObjectsInternals table.

```
PROCEDURE AddObjectValueText (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value nvarchar(max),
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to which the attribute is to be added. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute that is being added. This MUST NOT be NULL. This SHOULD be a valid Attribute Type Name, as specified in the tables in section [2.2.1.6], for the Object Type of the object that is specified by @ObjectID, and whose Attribute Data Type is Text.

@Value: The value of the attribute. If the object that is identified by @ObjectID already has an attribute identified by @AttributeName , and that attribute is Single-valued, and @Value matches the current value of the attribute, the stored procedure does nothing; otherwise, it constructs an attribute where the ValueText field is set to the value of this parameter. This MUST NOT be NULL.

@Cause: The Cause value to set for the AddCause field of the attribute. This MUST NOT be NULL.

@Time: A Time value to be set for the AddTime field of the attribute. If the value is NULL, the server MUST set the AddTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: MUST NOT return any result sets.

3.1.5.8 CreateManagementPolicyRule

The CreateManagementPolicyRule stored procedure is invoked to create a Management Policy Rule. The new Management Policy Rule is created by adding a row to the ObjectsInternal table for each attribute of the Management Policy Rule. The set of attributes for a Management Policy Rule is defined by the table in Management Policy Rule (section 3.1.1.1.7) and Common Attributes (section 2.2.1.6.1). For each attribute, the server MUST set the Object Type Key to 21 (Management Policy Rule) and the DeleteTime to "9999-01-01 00:00:00.000". Unless otherwise specified, the remaining fields of the attribute MUST be set to NULL. The ValueString field of the ObjectType attribute MUST be set to "ManagementPolicyRule", the ValueDateTime field of the DeletedTime attribute MUST be set to "9999-01-01 00:00:00.000", and the ValueDateTime field of the CreatedTime attribute MUST be set to the server's current time.

```
PROCEDURE CreateManagementPolicyRule (
@ObjectID uniqueidentifier = null,
@ActionParameter xml = null,
@ActionType xml = null,
@ActionWorkflowDefinition xml = null,
@AuthenticationWorkflowDefinition xml = null,
@AuthorizationWorkflowDefinition xml = null,
@Description nvarchar(448) = null,
@DisplayName nvarchar(448),
@GrantRight bit = 1,
@PrincipalSet uniqueidentifier = null,
@ResourceCurrentSet uniqueidentifier = null,
@ResourceFinalSet uniqueidentifier = null,
@Cause uniqueidentifier,
@Disabled bit = 0,
);
```

@ObjectID: The Object Identifier of the Management Policy Rule. The server MUST use this as the value for the ObjectID field for each attribute of the Management Policy Rule, as well as for the ValueReference field of the ObjectID attribute.

@ActionParameter: The ValueList of the Attribute Type Names that are used to match the Management Policy Rule to Requests, as described in section 3.1.1.1.3. This MUST be "<v>*</v>", which means all attributes. For each element of the @ActionParameter ValueList, the server MUST create an ActionParameter attribute for this Management Policy Rule and assign the element value to the ValueString field.

@ActionType: The ValueList of Rule Action Type that is used to match the Management Policy Rule to Requests, as described in section 3.1.7.1. This MUST be "<v>Read</v><v>Create</v> <v>Delete</v><v>Modify</v><v>Remove</v>". For each element of the @ActionType ValueList, the Server MUST create an ActionType attribute, where the element value is assigned to the ValueString field.

@ActionWorkflowDefinition: The Client MUST send empty string. This MUST be ignored by the Server.

@AuthenticationWorkflowDefinition: The Client MUST send empty string This MUST be ignored by the Server.

@AuthorizationWorkflowDefinition: The Client MUST send empty string This MUST be ignored by the Server.

@Description: The value for the ValueString field of the Description attribute of the Management Policy Rule. This MUST be default.

@DisplayName: The value for the ValueString field of the DisplayName attribute of the Management Policy Rule.

@GrantRight: The value for the ValueBoolean field of the GrantRight attribute of the Management Policy Rule. This MUST be default. This MUST be 1.

@PrincipalSet: The value for the ValueReference field of the PrincipalSet attribute of the Management Policy Rule.

@ResourceCurrentSet: The value for the ValueReference field of the ResourceCurrentSet attribute of the Management Policy Rule.

@ResourceFinalSet: The value for the ValueReference field of the ResourceFinalSet attribute of the Management Policy Rule.

@Cause: The Cause value to set for the AddCause field of each attribute of the Management Policy Rule. This MUST NOT be NULL.

@Disabled: The ValueBoolean field of the Disabled attribute of the Management Policy Rule. This MUST be 0.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.9 CreateMembershipCondition

The CreateMembershipCondition is invoked to create a Membership Condition that is used for defining the Set membership. The Membership Condition is created by adding a row in MembershipCondition table.

```
PROCEDURE CreateMembershipCondition (
@membershipCondition nvarchar(max),
@membershipConditionIdentifier uniqueidentifier OUTPUT,
);
```

@membershipCondition: The Membership Condition. The server MUST assign this value to the MembershipConditionStatement field of the created Membership Condition.

@membershipConditionIdentifier: A unique identifier for the created Membership Condition. The server MUST create a new unique identifier and assign it to @membershipConditionIdentifier.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.10 CreateMembershipConditionStatement

The CreateMembershipConditionStatement creates a Membership Condition Statement. The Membership Condition Statement is created by adding a row to the MembershipConditionStatement table.

```
PROCEDURE CreateMembershipConditionStatement (
@xpath nvarchar(max),
@referentIdentifier uniqueidentifier,
@attribute nvarchar(448),
@operatorEqual bit,
@operatorGreater bit,
@operatorLess bit,
@operatorLike bit,
@operatorInversion bit,
@literalValueBoolean bit = null,
@literalValueDateTime datetime = null,
@literalValueInteger bigint = null,
@literalValueReference uniqueidentifier = null,
@literalValueString nvarchar(max) = null,
@functionValue nvarchar(max) = null,
@missingValue bit = 0,
@dereferencedValueQuery nvarchar(max) = null,
@dereferencedValueMembershipConditionIdentifiers nvarchar(max) = null,
@dereferencedValueAttribute nvarchar(448) = null,
@recursivelyDereferencedAttribute nvarchar(448) = null,
@recursionFrom bit,
@recursionTo bit,
@attributes nvarchar(max),
@dereferencedExplicitMembershipSetIdentifier uniqueidentifier = null,
@statementIdentifier uniqueidentifier OUTPUT,
);
```

@xpath: The xpath that defines the condition statement. This MUST be "ObjectType= 'ma-data'" or "ObjectType='mv-data'".

@referentIdentifier: This MUST be "B5E4B9F5-FCF4-4BDD-8488-19B107BC6BB5". This MUST be ignored by the server.

@attribute: The Attribute Type Name. This MUST be "ObjectType". The server MUST assign this value to the Attribute field of the created MembershipConditionStatement

@operatorEqual: The value is 1 if the operator is equal; otherwise, the value is 0. This MUST be 1. The server MUST assign this value to OperatorEqual field of the created MembershipConditionStatement.

@operatorGreater: The client MUST send 0. This MUST be ignored by the server.

@operatorLess: The client MUST send 0. This MUST be ignored by the server.

@operatorLike: The client MUST send 0. This MUST be ignored by the server.

@operatorInversion: The client MUST send 0. This MUST be ignored by the server.

@literalValueBoolean: The client MUST send "default". This MUST be ignored by the server.

@literalValueDateTime: The client MUST send "default". This MUST be ignored by the server.

@literalValueInteger: The client MUST send "default". This MUST be ignored by the server.

@literalValueReference: The client MUST send "default". This MUST be ignored by the server.

@literalValueString: The Object Type Name. This MUST be "ma-data" or "mv-data". The server MUST assign the value to literalValueString field.

@functionValue: The client MUST send "default". This MUST be ignored by the server.

@missingValue: The client MUST send 0. This MUST be ignored by the server.

@dereferencedValueQuery: The client MUST send "default". This MUST be ignored by the server.

@dereferencedValueMembershipConditionIdentifiers: The client MUST send "default". This MUST be ignored by the server.

@dereferencedValueAttribute: The client MUST send "default". This MUST be ignored by the server.

@recursivelyDereferencedAttribute: The client MUST send "default". This MUST be ignored by the server.

@recursionFrom: The client MUST send 0. This MUST be ignored by the server.

@recursionTo: The client MUST send 0. This MUST be ignored by the server.

@attributes: The Membership Condition Statement Attribute. This MUST be '?<Statement><Attributes> <Attribute>ObjectType</Attribute> </Attributes></Statement>'. The server MUST store it on the Attributes field of the created MembershipConditionStatement.

@dereferencedExplicitMembershipSetIdentifier: The client MUST send "default". This MUST be ignored by the server.

@statementIdentifier: A unique identifier of the Membership Condition statement. The Server MUST create a new unique identifier and assign it to @statementIdentifier and to the StatementIdentifier field of the created MembershipConditionStatement.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.11 CreateRequest

The CreateRequest stored procedure is invoked to create a new Resource that has an Object Type Name value of a Request. The new Request is created by adding a row to the ObjectsInternal table for each attribute of the Request. The set of attributes for a Request is defined by the table in Request (section 3.1.1.1.3) attributes and Common Attributes (section 2.2.1.6.1). For each attribute, the server MUST set the Object Type Key to 26 (Request) and the DeleteTime to "9999-01-01 00:00:00.000". Unless otherwise specified as follows, the remaining fields of the attribute MUST be set to NULL. The ValueString field of the ObjectType attribute MUST be set to Request, the ValueDateTime field of the DeletedTime attribute MUST be set to "9999-01-01 00:00:00.000", the ValueDateTime field of the CreatedTime attribute MUST be set to the server's current time, and the ValueString field of the RequestStatus to "validating".

```
PROCEDURE CreateRequest (
    @targetIdentifier uniqueidentifier = null,
    @creator uniqueidentifier,
    @anonymous bit,
    @operation nvarchar(448),
    @parameters xml = null,
    @ancillaryParameters xml = null,
    @displayNameTemplate nvarchar(448),
    @locale nvarchar(448) = null,
    @cause uniqueidentifier,
    @isChildRequest bit = null,
    @requestIdentifier uniqueidentifier OUTPUT,
    @displayName nvarchar(448),
    @targetObjectTypeName nvarchar(448) OUTPUT,
    @time datetime OUTPUT,
);
```

@targetIdentifier: The Target Object Identifier. The server MUST use this value to set the ValueReference field of the Target attribute for this Request. If @operation is 'Create', then the server MUST generate a new Object Identifier for the new Resource.

@creator: The Cause that creates the Request. This MUST NOT be NULL. The server MUST use this value to set the ValueReference field of the Creator attribute.

@anonymous: The client MUST send 0. This MUST be ignored by the server.

@operation: The Request Operation. This MUST NOT be NULL. The server MUST use this value to set the ValueString field of the Operation attribute of the created Request.

@parameters: The Request Parameters. The server MUST use this value to set the valueText field of the RequestParameter attribute of the created Request.

@ancillaryParameters: The client MUST send "default". This MUST be ignored by the server.

@displayNameTemplate: The request display name template. This MUST be one of the values from the table in Request Display Name Template (section 2.2.1.18). The server MUST use this value for setting the ValueString field of the DisplayName attribute. The server MUST replace {0} by the ValueString field of the Object Type Name attribute of the Target Resource. The server MUST replace {1} with the ValueString field of the DisplayName attribute of the Target Resource.

@locale: The client MUST send default. This MUST be ignored by the server.

@cause: The Cause. The server MUST set the AddCause field for each attribute of the Request to this value.

@isChildRequest: The Client MUST send 0. This MUST be ignored by server.

@requestIdentifier: The created Request Object Identifier. The server MUST create a new unique identifier for the Request, assign it to the ValueReference field of the ObjectID attribute, and assign it to @requestUdentifier.

@displayName: The client MUST send NULL. The server MUST assign the value of the valueString field of the DisplayName attribute to @displayName.

@targetObjectTypeName: The Object Type Name of the Target Resource. The server MUST assign the ValueString field of the TargetObjectType attribute to the @targetObjectTypeName.

@time: The time at which the Request is created. The server MUST assign @time the value of the ValueDateTime field of the CreatedTime attribute for this Request.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.12 DeleteCachedSet

DeleteCachedSet stored procedure is invoked to delete a Set that has a non-NULL Filter attribute. This will set the DeleteTime and DeleteCause of all non-deleted attributes of the Set in the ObjectsInternal table. This stored procedure MUST evaluate the other Sets.

```
PROCEDURE DeleteCachedSet (
@ObjectID uniqueidentifier,
@Cause uniqueidentifier
);
```

@ObjectID: The Object Identifier of the Set to be deleted. This MUST NOT be NULL.

@Cause: The Cause value to set for the DeleteCause field of the attributes of the Set. This MUST NOT be NULL.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.13 DeleteManagementPolicyRule

The DeleteManagementPolicyRule stored procedure is invoked to set DeleteTime and DeleteCause on all non-deleted attributes in the ObjectsInternal table where the attributes ObjectID matches the specified Object Identifier and @ObjectID refers to a Management Policy Rule. This call MUST call Set Transitions, as described in section 3.1.7.4.

```
PROCEDURE DeleteManagementPolicyRule (
@ObjectID uniqueidentifier,
@Cause uniqueidentifier,
@Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Management Policy Rule to be deleted. This MUST NOT be NULL.

@Cause: The Cause value to set for the DeleteCause field of the attributes of the Management Policy Rule. This MUST NOT be NULL.

@Time: A Time value to be set for the DeleteTime field of the attributes of the Management Policy Rule. If NULL, the server MUST set the DeleteTime of the attributes to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.14 DeleteObject

The DeleteObject stored procedure is invoked to set the DeleteTime and DeleteCause of all non-deleted attributes in the ObjectsInternal table where the attributes ObjectID matches the specified Object Identifier, thereby deleting the object.

```
PROCEDURE DeleteObject (
    @ObjectID uniqueidentifier,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource to be deleted. This MUST NOT be NULL.

@Cause: The Cause value to set for the DeleteCause field of the attributes of the Resource. This MUST NOT be NULL.

@Time: A Time value to be set for the DeleteTime field of the attributes of the Resource. If the value is NULL, the server MUST set the DeleteTime of the attributes to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.15 DeleteObjectValueBinary

The DeleteObjectValueBinary stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is Binary, thereby deleting all matching attributes from the given object Resource.

```
PROCEDURE DeleteObjectValueBinary (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value varbinary(max),
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueBinary field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time value to be assigned to the DeleteTime of the attribute. If the value is NULL, the server MUST set the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.16 DeleteObjectValueBoolean

The DeleteObjectValueBoolean stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is Boolean, thereby deleting all matching attributes from the given object Resource.

```
PROCEDURE DeleteObjectValueBoolean (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value bit,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueBoolean field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time value to be assigned to the DeleteTime of the attribute. If NULL, the server MUST set the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.17 DeleteObjectValueDateTime

The DeleteObjectValueDateTime stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is DateTime, thereby deleting all matching attributes from the given object Resource.

```
PROCEDURE DeleteObjectValueDateTime (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value datetime,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueDateTime field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time to be assigned to the DeleteTime of the attribute value. If the value is NULL, the server will assign the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.18 DeleteObjectValueInteger

The DeleteObjectValueInteger stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is Integer, thereby deleting all matching attributes from the given object Resource.

```
PROCEDURE DeleteObjectValueInteger (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value bigint,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueInteger field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time to be assigned to the DeleteTime of the attribute value. If the value is NULL, the server will assign the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.19 DeleteObjectValueReference

The DeleteObjectValueReference stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is Reference, thereby deleting all matching attributes from the given object Resource.

```
PROCEDURE DeleteObjectValueReference (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value uniqueidentifier,
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueReference field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time to be assigned to the DeleteTime of the attribute value. If the value is NULL, the server will assign the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.20 DeleteObjectValueString

The DeleteObjectValueString stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is a String, thereby deleting all matching attributes from the given object Resource.

```
PROCEDURE DeleteObjectValueString (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value nvarchar(448),
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueString field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time to be assigned to the DeleteTime of the attribute value. If the value is NULL, the server will assign the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.21 DeleteObjectValueText

The DeleteObjectValueText stored procedure is invoked to set the DeleteTime and DeleteCause of an attribute in the ObjectsInternal table, where the Attribute Data Type of the attribute is Text, thereby deleting all matching attributes from the given object resource.

```
PROCEDURE DeleteObjectValueText (
    @ObjectID uniqueidentifier,
    @AttributeName nvarchar(448),
    @Value nvarchar(max),
    @Cause uniqueidentifier,
    @Time datetime = null,
);
```

@ObjectID: The Object Identifier of the resource from which the attribute is to be deleted. This MUST NOT be NULL.

@AttributeName: The Attribute Type Name of the attribute to be deleted. This MUST NOT be NULL. If the attribute does not exist for the object, the server does nothing.

@Value: The current value of the attribute to be deleted. This MUST NOT be NULL. If this does not match the valueText field of any matching attribute, the server does nothing.

@Cause: The Cause to be assigned to the DeleteCause of the attribute. This MUST NOT be NULL.

@Time: The Time to be assigned to the DeleteTime of the attribute value. If the value is NULL, the server will assign the DeleteTime of the attribute to the server's current time.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.22 DequeueWorkItem

The DequeueWorkItem is invoked to retrieve a scheduled item from the workflow work queue.

```
PROCEDURE DequeueWorkItem (
    @hostId uniqueidentifier,
    @lockTimeout int,
    @result int OUTPUT,
);
```

@hostId: This MUST be ignored by the server.

@lockTimeout: This MUST be ignored by the server.

@result: Result code. This MUST be 1.

Return Values: An integer that MUST be 0.

Result Sets: This stored procedure MUST return a DequeueWorkItem.ResultSet0.

3.1.5.23 EvaluateNewRequest

The EvaluateNewRequest stored procedure is invoked to create a new Request that has an Object Type Name value of the Request. The new Request is created by adding a row to the ObjectsInternal table for each attribute of the Request. The set of attributes for a Request is defined by the table in Request attributes (section 2.2.1.6.9) and Common Attributes (section 2.2.1.6.1). For each attribute, the server MUST set the Object Type Key to 26 (Request) and the DeleteTime to "9999-01-01 00:00:00.000". Unless otherwise specified as follows, the remaining fields of the attribute MUST be set to NULL. The ValueString field of the ObjectType attribute MUST be set to Request, the ValueDateTime field of the DeletedTime attribute MUST be set to "9999-01-01 00:00:00.000", and the ValueDateTime field of the CreatedTime attribute MUST be set to the server's current time. The server MUST call the Permission Evaluation local event, as described in section 3.1.7.2 . The stored procedure MUST call the Business Rule Evaluation local event, as described in section 3.1.7.6. and the stored procedure MUST set the ValueString field of the RequestStatus as either "Validated" or "Denied".

```
PROCEDURE EvaluateNewRequest (
    @targetIdentifier uniqueidentifier = null,
    @creator uniqueidentifier,
    @anonymous bit,
    @operation nvarchar(448),
    @parameters xml = null,
    @ancillaryParameters xml = null,
    @displayNameTemplate nvarchar(448),
    @locale nvarchar(448) = null,
    @cause uniqueidentifier,
    @isChildRequest bit = null,
    @requestIdentifier uniqueidentifier OUTPUT,
    @displayName nvarchar(448) OUTPUT,
    @targetObjectTypeName nvarchar(448) OUTPUT,
    @time datetime OUTPUT,
    @exception nvarchar(max) OUTPUT,
);
```

@targetIdentifier: The Target Object Identifier. The server MUST use this value to set the ValueReference field of the Target attribute for this Request. In case of a create Request, the server MUST generate a new **GUID** for the new Resource.

@creator: The Cause that creates the Request. This MUST NOT be NULL. The server MUST use this value to set the ValueReference field of the Creator attribute.

@anonymous: This Client MUST send 0. This MUST be ignored by the server.

@operation: The Request Operation. This MUST NOT be NULL. The server MUST use this value to set the ValueString field of the Operation attribute of the created Request.

@parameters: The Request Parameters. The server MUST use this value to set the valueText field of the RequestParameter attribute of the created Request.

@ancillaryParameters: The client MUST "send default". This MUST be ignored by the server.

@displayNameTemplate: The request display name template. This MUST be one of the values from the table in Request Display Name Template (section 2.2.1.16). The server MUST use this value for setting the ValueString field of the DisplayName attribute. The server MUST replace {0} by the Object Type Name of the Target Resource. The server MUST replace {1} with the ValueString of the DisplayName attribute of the Target Resource.

@locale: The client MUST send default. This MUST be ignored by the server.

@cause: The Cause. The server MUST set the AddCause field for each attribute of the Request to this value.

@isChildRequest: The Client MUST send 0. This MUST be ignored by server.

@requestIdentifier: The created Request Object Identifier. The server MUST create a new unique identifier for the Request, assign it to the ValueReference field of the ObjectID attribute or the Request, and assign it to @requestUdentifier.

@displayName: The client MUST send NULL. The server MUST assign the value of the valueString field of the DisplayName attribute to @displayName.

@targetObjectTypeName: The Object Type Name of the Target Resource. The server MUST assign the ValueString field of the TargetObjectType attribute to the @targetObjectTypeName.

@time: The Time at which the Request is created. The server MUST assign @time the value of the ValueDateTime field of the CreatedTime attribute for this Request.

@exception: The message that describes the invalid request exception if an exception was generated by the sproc.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return an EvaluateNewRequest.ResultSet0. The server MUST call the Workflow Instance Creation local event, as described in section 3.1.7.5. If there is no Workflow Instance created, the server MUST return no rows in the result set. Otherwise, the server MUST assign the valueReference field of the Object Identifier attribute of the Management Policy Rule to the RuleIdentifier field and valueReference field of the Object Identifier attribute of the created Workflow Instance to ActionWorkflowInstanceIdentifier field.

This stored procedure MUST return an EvaluateNewRequest.ResultSet1. The server MUST call the Set Transition Evaluation local event, as described in section 3.1.7.3, by passing in the Object Identifier of the created Request and the @cause and @Time parameter values. For each Set Transition element returned by the local event, the server MUST assign the fields of the SetTransition element to the same-named fields of the result set.

3.1.5.24 EvaluateRequest

The EvaluateRequest stored procedure is invoked to evaluate a given Request. This server MUST call the Permission Evaluation local event, as described in section 3.1.7.2. The server MUST call the Business Rule Evaluation local event, as described in section 3.1.7.6, and the server MUST set the ValueString field of the RequestStatus as either "Validated" or "Denied".

```
PROCEDURE EvaluateRequest (
    @requestIdentifier uniqueidentifier,
    @doNotEnforceRights bit = 0,
    @bypassAuthentication bit = 0,
    @bypassAuthorization bit = 0,
    @bypassAction bit = 0,
    @suppressOutput bit = 0,
    @processingTime datetime = null,
    @requestLocked bit = 0,
);
```

@requestIdentifier: The Request Object Identifier of the Request to be evaluated. This MUST NOT be NULL.

@doNotEnforceRights: The client MUST send 0. This MUST be ignored by the server.

@bypassAuthentication: The client MUST send 0. This MUST be ignored by the server.

@bypassAuthorization: The client MUST send 0. This MUST be ignored by the server.

@bypassAction: The client MUST send 0. This MUST be ignored by the server.

@suppressOutput: The client MUST send 0. This MUST be ignored by the server.

@requestLocked: The client MUST send 0. This MUST be ignored by the server.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return an EvaluateRequest.ResultSet0. The server MUST call the Workflow Instance Creation local event, as described in section 3.1.7.5. If no Workflow Instance is created, the server MUST not return any rows in the result set. Otherwise, the server MUST assign the valueReference field of the Object identifier attribute of the Management Policy Rule to the RuleIdentifier field and assign the valueReference field of the Object Identifier attribute of the created Workflow Instance to the ActionWorkflowInstanceIdentifier field.

This stored procedure MUST return an EvaluateRequest.ResultSet1. The server MUST call the Set Transition Evaluation local event, as described in section 3.1.7.3, by passing in the Object Identifier of the created Request and the @cause and @Time parameter values. For each Set Transition element returned by the local event, the server MUST assign the fields of the SetTransition element to the same-named fields of the result set.

3.1.5.25 GetObjectCurrent

The GetObjectCurrent stored procedure is invoked to retrieve all non-deleted attributes for the specified object resource. All entries in the result sets have the same ObjectID value.

```
PROCEDURE GetObjectCurrent (
    @ObjectID uniqueidentifier,
    @Locale nvarchar(448) = null,
    @AttributeNames xml = null,
);
```

@ObjectID: The Object Identifier of the object whose attributes are to be retrieved. This MUST NOT be Null.

@Locale: The client MUST send NULL. This MUST be ignored by the server.

@AttributeNames: This MUST be ignored by the server.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a GetObjectCurrent.ResultSet0, which contains the Object Type Name of the object that is identified by @ObjectID.

This stored procedure MUST return a GetObjectCurrent.ResultSet1, which contains an entry for each attribute where ObjectId is @ObjectId and where DeletedCause is NULL.

3.1.5.26 GetObjectCurrentRelative

The GetObjectCurrentRelative stored procedure is invoked to retrieve attributes for the specified object resource. Only attributes for which the specified principal has read access are returned. All entries in the result set have the same ObjectID value.

```
PROCEDURE GetObjectCurrentRelative (
    @ObjectID uniqueidentifier,
    @Principal uniqueidentifier,
    @Locale NVARCHAR(448) = null,
    @AttributeNames xml = null,
);
```

@ObjectID: The Object Identifier of the object whose attributes are to be retrieved. This MUST NOT be NULL.

@Principal: The Object Identifier of the principal. This MUST NOT be NULL.

@Locale: The client MUST send NULL. This MUST be ignored by the server.

@AttributeNames: The ValueList of the Attribute Type Names of the attributes to retrieve. If the value is NULL, all attributes of that Resource are to be retrieved.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a GetObjectCurrentRelative.ResultSet0, which contains the Object Type Name of the object that is identified by @ObjectID.

This stored procedure MUST return a GetObjectCurrentRelative.ResultSet1, which contains an entry for each attribute where ObjectID is @ObjectID, DeletedCause is NULL, and for which the principal specified in @Principal is authorized for read access.

3.1.5.27 GetObjectType

The GetObjectType stored procedure is invoked to retrieve the Object Type Name of the specified object resource.

```
PROCEDURE GetObjectType (
@ObjectID uniqueidentifier,
@ObjectType nvarchar(max) OUTPUT,
);
```

@ObjectID: The Object Identifier of the object whose Object Type Name is to be retrieved. This MUST NOT be Null.

@ObjectType: The Object Type Name of the object resource specified by @ObjectID.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.28 GetPersonRegistrationsAll

The GetPersonRegistrationsAll stored procedure does nothing.

```
PROCEDURE GetPersonRegistrationsAll (
@objectId uniqueidentifier
);
```

@objectId: The Object Identifier of a Person. The server MUST ignore this.

Return Values: This MUST NOT return any values.

Result Sets: This MUST NOT return any result sets.

3.1.5.29 InsertInstance

The InsertInstance is invoked by the Durable Services SQL persistence provider to insert a new durable code service instance in the database.

```
PROCEDURE InsertInstance (
    @id uniqueidentifier,
    @instance image = null,
    @instanceXml xml = null,
    @unlockInstance bit,
    @hostId uniqueidentifier,
    @lockTimeout int,
    @result int OUTPUT,
);
```

@id: The identifier of the durable code service instance.

@instance: This MUST be NULL.

@instanceXml: The serialized XML of the durable code service instance.

@unlockInstance: 1 to unlock the durable code service instance; otherwise, 0.

@hostId: The lock owner identifier of the host that is calling the stored procedure. This MUST be NULL if the locking is not enabled.

@lockTimeout: The lock expiry timeout. This MUST be NULL if the locking is not enabled.

@result: The execution result of the stored procedure.

Value	Description
0	Success.
1	Instance already exists.
2	Unknown error.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.30 InsertInstanceState

The InsertInstanceState stored procedure saves a new workflow instance or updates deletes an existing one.

```
PROCEDURE InsertInstanceState (
    @uidInstanceID uniqueidentifier,
    @state image,
    @status int,
    @unlocked int,
    @blocked int,
    @info ntext,
    @ownerID uniqueidentifier = null,
    @ownedUntil datetime = null,
    @nextTimer datetime,
    @result int OUTPUT,
    @currentOwnerID uniqueidentifier OUTPUT,
);
```

@uidInstanceID: The identifier of the workflow instance. This MUST NOT Be NULL.

@state: This MUST be 0x.

@status: The serialized binary large object (BLOB) of the workflow instance. This MUST NOT be NULL.

@unlocked: Indicates whether to unlock the instance or not. This MUST NOT be NULL.

@blocked: Indicates whether this workflow instance is blocked and it can proceed with work or is blocked on a timer or an external event. This MUST NOT be NULL.

@info: The suspension or termination reason of the workflow instance.

@ownerID: The lock owner identifier of the host calling the stored procedure. It MUST be NULL if locking is not enabled.

@ownedUntil: The lock expiry timeout. It MUST be NULL if locking is not enabled.

@nextTimer: The lock expiry timeout. It MUST be NULL if locking is not enabled.

@result: The execution result of the stored procedure.

Value	Description
-2	Instance was not locked by the owner trying to update it (lock may have timed out).
0	Stored procedure executed successfully.

@currentOwnerID: The identifier of the caller if the caller was the owner of the workflow instance; otherwise, the actual owner identifier of the workflow instance. This MUST be NULL if locking is not enabled.

Return Values: This MUST NOT return any values.

Result Sets: This MUST NOT return any result sets.

3.1.5.31 PostProcessUpdate

The PostProcessUpdate MUST call the Set Transition local event for the Request, as described in section 3.1.7.4.

```
PROCEDURE PostProcessUpdate (
    @resourceIdentifier uniqueidentifier,
    @cause uniqueidentifier,
    @time datetime,
);
```

@resourceIdentifier: The Target Object Identifier. This MUST NOT be NULL.

@cause: The Request Object Identifier. This MUST NOT be NULL.

@time: The Time that server MUST use to set the AddCause field for the ComputedMembers attribute. This MUST NOT be NULL.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.32 PreProcessUpdate

The PreProcessUpdate calls the Business Rule Evaluation local event, as described in section 3.1.7.6. If there is a violation, the stored procedure MUST raise an exception.

```
PROCEDURE PreProcessUpdate (
    @resourceIdentifier uniqueidentifier,
    @cause uniqueidentifier,
    @requestLocked bit = 0,
);
```

@resourceIdentifier: The Target Object Identifier to which the Business Rules are to be applied. This MUST NOT be Null.

@cause: The Cause, which is the Request Object Identifier that caused this change. This MUST NOT be Null.

@requestLocked: This MUST BE 0.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.33 ProcessRequest

The ProcessRequest stored procedure is invoked to process a given Request. The server MUST call the Evaluate Business Rules local event, as described in section 3.1.7.5, then it MUST update ObjectsInternal with the effect of the RequestParameters. Subsequently, it MUST call the Set Transition local event for the Request, as described in section 3.1.7.4, and then it MUST set the valueString field of the RequestStatus attribute to "Committed".

```
PROCEDURE ProcessRequest (
    @identifier uniqueidentifier,
    @resourceIdentifier uniqueidentifier OUTPUT,
);
```

@identifier: The Request Object Identifier of the Request to be processed.

@resourceIdentifier: The Target Object Identifier. The server MUST set this to the ValueReference field of the Target attribute of the Request identified by @identifier.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.34 RefreshCurrentObjectCache

The RefreshCurrentObjectCache stored procedure does nothing.

```
PROCEDURE RefreshCurrentObjectCache (
);
```

Return Values: An integer that MUST be 0.

Result Sets: this MUST NOT return any result sets.

3.1.5.35 RefreshPolicyCaches

The RefreshPolicyCaches stored procedure does nothing.

```
PROCEDURE RefreshPolicyCaches (
);
```

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.36 RegisterService

The RegisterService stored procedure is invoked to return the Host Identifier of a protocol client.

```
PROCEDURE RegisterService (
@hostName nvarchar(128),
@hostId smallint OUTPUT,
);
```

@hostName: The host machine name of a protocol client.

@hostId: The Host Identifier. If the @hostName matches any of the hostname field values on RegisteredService table, the server MUST set @hostId by the HostId field value. Otherwise, the server MUST generate a new unique HostId and create an entry in the RegisteredService table by the new HostId and @hostName. It MUST set @hostId to the HostId field value of that entry.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.37 RetrieveExpiredTimerIds

The RetrieveExpiredTimerIds is invoked to retrieve workflow instance identifiers with expired timers.

```
PROCEDURE RetrieveExpiredTimerIds (
    @ownerID uniqueidentifier = null,
    @ownedUntil datetime = null,
    @now datetime,
);
```

@ownerID: This MUST be ignored.

@ownedUntil: This MUST be ignored.

@now: Current time. Expired timers are timers with an expiry time that is less than the current time.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a RetrieveExpiredTimerIds.ResultSet0 that contains an entry for each workflow instance whose expiredTime is older than @now.

3.1.5.38 RetrieveWorkflowDataForHostActivator

The RetrieveWorkflowDataForHostActivator stored procedure is invoked to retrieve Request and workflow information. The client calls this store procedure to retrieve the workflow that is required to be run.

```
PROCEDURE RetrieveWorkflowDataForHostActivator (
    @hostId smallint,
    @pingIntervalSecs smallint,
    @activeHostedWorkflowDefinitionSequenceNumber int,
    @workflowControlMessagesMaxPerMinute smallint = 60,
    @RequestRecoveryMaxPerMinute smallint = 30,
    @RequestCleanupMaxPerMinute smallint = 120,
    @runRequestRecoveryScan bit = 0,
    @doPolicyApplicationDispatch bit OUTPUT
);
```

@hostId: The Host Identifier that the client received through a previous a call to RegisterService.

@pingIntervalSecs: The time delay period, in seconds, between calls that the client expects to make to this procedure.

@activeHostedWorkflowDefinitionSequenceNumber: The Workflow Definition sequence number with which to begin the enumeration of Workflow Definitions. This MUST be 0. A value of 0 retrieves all Workflow Definitions.

@workflowControlMessagesMaxPerMinute: The client MUST send 60 and this MUST be ignored by the server.

@RequestRecoveryMaxPerMinute: The client MUST send 30 and this MUST be ignored by the server.

@RequestCleanupMaxPerMinute: The client MUST send 120 and this MUST be ignored by the server.

@runRequestRecoveryScan: The client MUST send 0 and this MUST be ignored by the server.

@doPolicyApplicationDispatch: The server MUST set this to 0 and it MUST be ignored by the client.

Return Values: An integer that MUST be 0.

Result Sets: This stored procedure MUST return a RetrieveWorkflowDataForHostActivator.ResultSet0. The server MUST return a row for each Workflow Definition, starting with the Workflow Definition whose sequence number is @activeHostedWorkflowDefinitionSequenceNumber.

This stored procedure MUST return an empty RetrieveWorkflowDataForHostActivator.ResultSet1.

This stored procedure MUST return an empty RetrieveWorkflowDataForHostActivator.ResultSet2.

3.1.5.39 RetrieveWorkflowInstance

The RetrieveWorkflowInstance stored procedure is invoked to retrieve information about a Workflow Instance.

```
PROCEDURE RetrieveWorkflowInstance (
    @identifier uniqueidentifier,
);
```

@identifier: The Object Identifier of the Workflow Instance whose information is to be retrieved. This MUST NOT be NULL.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return a RetrieveWorkflowInstance.ResultSet0, in which the DefinitionIdentifier is set to the value of the ValueReference field of the Workflow Definition attribute of the Workflow Instance, the RequestIdentifier is set to the value of the ValueReference field of the Request attribute of the Workflow Instance, the RequestorIdentifier is set to the value of the ValueReference field of the Requestor attribute of the Workflow Instance, the TargetIdentifier is set to the value of the ValueReference field of the Target attribute of the Workflow Instance, and Status is set to the value of the ValueInteger field of the WorkflowStatus attribute of the Workflow Instance.

3.1.5.40 RetrieveWorkflowDefinition

The RetrieveWorkflowDefinition stored procedure is invoked to retrieve information about a Workflow Definition.

```
PROCEDURE RetrieveWorkflowInstanceDefinition (
    @workflowInstanceIdentifier uniqueidentifier,
    @workflowDefinitionIdentifier uniqueidentifier OUTPUT,
    @xoml nvarchar(max) OUTPUT,
    @rules nvarchar(max) OUTPUT,
    @deleteCause uniqueidentifier OUTPUT,
);
```

@workflowInstanceIdentifier: The Object Identifier of the Workflow Instance whose Workflow Definition is to be retrieved.

@workflowDefinitionIdentifier: The Object Identifier of the Workflow Definition that was used to create the Workflow Instance that is identified by @workflowInstanceIdentifier. The server MUST set this to the value of the valueReference field of the WorkflowDefinition attribute of the Workflow Instance.

@xoml: The Workflow XOML. The server MUST set this to the value of the valueString field of the XOML attribute of the Workflow Definition identified by @workflowDefinitionIdentifier.

@rules: The server MUST return NULL. It MUST be ignored by the client.

@deleteCause: The server MUST set this to the value of the valueReference field of the deleteCause attribute of the Workflow Definition, if the Workflow Definition has been deleted; otherwise, the value is NULL.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.41 UpdateRequest

The UpdateRequest stored procedure is invoked to update the RequestStatus and ActionWorkflowInstance attributes of a Request. The server MUST call the Set Transition local event for the Request, as described in section 3.1.7.4.

```
PROCEDURE UpdateRequest (
    @identifier uniqueidentifier,
    @targetIdentifier uniqueidentifier = null,
    @displayName nvarchar(448) = null,
    @description nvarchar(max) = null,
    @parameters xml = null,
    @authenticationWorkflowInstances xml = null,
    @authorizationWorkflowInstances xml = null,
    @actionWorkflowInstances xml = null,
    @committedTime datetime = null,
    @computedActors xml = null,
    @managementPolicies xml = null,
    @statusKey smallint = null,
    @status nvarchar(448) = null,
    @statusDetails xml = null,
    @cause uniqueidentifier,
    @extendedOutput tinyint = 1,
    @exception nvarchar(max) OUTPUT,
);
```

@identifier: The Request Object Identifier of the Request to be updated. This MUST NOT be NULL.

@targetIdentifier: The Target Object Identifier.

@displayName: The client MUST send default. This MUST be ignored by the server.

@description: The client MUST send default. This MUST be ignored by the server.

@parameters: The client MUST send default. This MUST be ignored by the server.

@authenticationWorkflowInstances: The client MUST send default or "<WorkflowInstances/>". This MUST be ignored by the server.

@authorizationWorkflowInstances: The client MUST send default or <WorkflowInstances/>. This MUST be ignored by the server.

@actionWorkflowInstances: A WorkflowInstances element . For each <Identifier> element within the <WorkflowInstances> element, the server MUST create an ActionWorkflowInstance attribute and assign the element value to its ValueReference field.

@committedTime: The Time that server MUST assign to the AddTime fields of the attributes that are created, or updated by this stored procedure and to the DeleteTime field of any attributes that was deleted during processing of this stored procedure. If NULL, the server MUST use the server's current time.

@computedActors: The client MUST send default. This MUST be ignored by the server.

@managementPolicies: The client MUST send default. This MUST be ignored by the server.

@statusKey: The Request Status Key for the Request. The server MUST convert the Request Status Key to the equivalent Request Status, as defined by being in the same row of the table in Request Status Key (section 2.2.1.11). The server MUST assign the resulting value to the ValueString field of the RequestStatus attribute. If the value is "Committed", the server MUST add a CommittedTime attribute to the Request by assigning the ValueDateTime field the value of @committedTime.

@status: The Request Status for the Request. If @statusKey is not NULL, the server MUST ignore this parameter; otherwise, the server MUST set the ValueString field of the RequestStatus attribute of the Request identified by @identifier to this value. If the value is "Committed", the server MUST add a CommittedTime attribute to the Request by assigning the ValueDateTime field the value of @committedTime.

@statusDetails: The client MUST send default. This MUST be ignored by the server.

@cause: The Cause that MUST be assigned to the AddCause field of any attribute that was added to the Request or updated during processing of this stored procedure, and to the DeleteCause field of any attribute of the Request that was deleted during processing of this stored procedure. This MUST NOT be NULL.

@extendedOutput: The client MUST send default. This MUST be ignored by the server.

@exception: If an exception occurred during processing of the call, this contains the message of the exception; otherwise, it is NULL.

Return Values: An integer that MUST be 0.

Result Sets:

This stored procedure MUST return an UpdateRequest.ResultSet0. The server MUST set reEvaluation to 0 and requestDependencyKey to NULL.

This stored procedure MUST return an UpdateRequest.ResultSet1. This result set MUST be empty.

This stored procedure MUST return an UpdateRequest.ResultSet2. The server MUST call the Workflow Instance Creation local event, as described in section 3.1.7.4. If no WorkflowInstances are created, the server MUST not return any rows on the result set. Otherwise, the server MUST assign the valueReference of the Object Identifier attribute of the created Workflow Instance to the WorkflowInstanceIdentifier field.

This stored procedure MUST return an empty UpdateRequest.ResultSet3.

3.1.5.42 UpdateManagementPolicyRule

The UpdateManagementPolicyRule stored procedure is invoked to update the PrincipalSet, ResourceCurrentSet, and ResouceFinalSet of a Management Policy Rule.

```
PROCEDURE UpdateManagementPolicyRule (
@ObjectID uniqueidentifier,
@PrincipalSetReplace bit = 0,
@PrincipalSet uniqueidentifier = '0',
@PrincipalRelativeToResourceReplace bit = 0,
@PrincipalRelativeToResource nvarchar(448) = '0',
@ResourceCurrentSetReplace bit = 0,
@ResourceCurrentSet uniqueidentifier = '0',
@ResourceCurrentRelativeToPrincipalReplace bit = 0,
@ResourceCurrentRelativeToPrincipal nvarchar(448) = '0',
@ResourceFinalSetReplace bit = 0,
@ResourceFinalSet uniqueidentifier = '0',
@ResourceFinalRelativeToPrincipalReplace bit = 0,
@ResourceFinalRelativeToPrincipal nvarchar(448) = '0',
@ActionParameterUpdateType nvarchar(16) = 'none',
@ActionParameter xml = 'none',
@AddedActionType xml = null,
@RemovedActionType xml = null,
@GrantRight bit = null,
@Disabled bit = null,
@Cause uniqueidentifier,
@Time datetime = null,
);
```

@ObjectID: The Object Identifier of the Management Policy Rule to be updated. The server MUST use this as the value for the ObjectID field for each attribute of the Management Policy Rule, as well as for the ValueReference field of the ObjectID attribute.

@PrincipalSetReplace: Specifies whether the PrincipalSet attribute of the Management Policy Rule is to be updated. A value of 1 indicates that the PrincipalSet attribute is to be updated. This MUST be 1.

@PrincipalSet: The ObjectIdentifier of PrincipalSet. If the @PrincipalSetReplace is 1, the server MUST assign this value to the valueReference field of the PrincipalSet attribute; otherwise, the server MUST ignore this.

@PrincipalRelativeToResourceReplace: The client MUST send 1. This MUST be ignored by the server.

@ResourceCurrentSetReplace: Specifies whether the ResourceCurrentSet attribute of the Management Policy Rule is to be updated. A value of 1 indicates that the ResourceCurrentSet attribute is to be updated. This MUST be 1.

@ResourceCurrentSet: The ObjectIdentifier of the ResouceCurrentSet. If the @ResouceCurrentSetReplace is 1, the server MUST assign this value to the valueReference field of the ResouceCurrentSet attribute; otherwise, the server MUST ignore this

@ResourceCurrentRelativeToPrincipalReplace: The client MUST send 1. This MUST be ignored by the server.

@ResourceFinalSetReplace: Specifies whether the ResourceFinalSet attribute of the Management Policy Rule is to be updated. A value of 1 indicates that the ResourceFinalSet attribute is to be updated. This MUST be 1.

@ResourceFinalSet: The ObjectIdentifier of the ResouceFinalSet. If the @ResouceFinalSetReplace is 1, the server MUST assign this value to the valueReference field of the ResouceFinalSet attribute; otherwise, the server MUST ignore this.

@ResourceFinalRelativeToPrincipalReplace: The client MUST send 1. This MUST be ignored by the server.

@Cause: The Cause value to set for the AddCause field of each attribute of the Management Policy Rule that is updated by this stored procedure. This MUST NOT be NULL.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.43 UpdateRequestStatusForCompletedActionWorkflows

The UpdateRequestStatusForCompletedActionWorkflows stored procedure is invoked to update the RequestStatus attribute to Completed for Requests that have ActionWorkflowInstance attributes that have been completed.

```
PROCEDURE UpdateRequestStatusForCompletedActionWorkflows (
    @maxRequestsToProcess smallint = 500
);
```

@maxRequestsToProcess: The maximum number of Requests to be updated in this invocation of the stored procedure.

Return Values: An integer that MUST be 0.

Result Sets: The stored procedure MUST return an UpdateRequestStatusForCompletedActionWorkflows.ResultSet0 that contains an entry for each Request whose Request Status was updated to Completed during the processing of this stored procedure. For each entry, the server MUST set the RequestIdentifier field to the value of the ValueReference field of the ObjectIdentifier of the Request, and MUST set the RequestStatus field to the value of the ValueString of the RequestStatus attribute of the Request.

The stored procedure MUST return an empty UpdateRequestStatusForCompletedActionWorkflows.ResultSet1.

The stored procedure MUST return an UpdateRequestStatusForCompletedActionWorkflows.ResultSet2 that contains an entry for each Request that the stored procedure tried to update but failed in doing so. The server MUST set the RequestIdentifier field of the entry to the value of the ValueString field of the Request.

3.1.5.44 UpdateSet

The Update Set stored procedure is invoked to process a Request for updating the set. This stored procedure calls the ProcessRequest stored procedure by passing @cause as the Request Object Identifier.

```
PROCEDURE UpdateSet (
    @setIdentifier uniqueidentifier,
    @cause uniqueidentifier,
    @Exception nvarchar(max) OUTPUT,
);
```

@setIdentifier: The Object Identifier of the Set that is being updated.

@cause: The Request Object Identifier of the Request that is updating the Set identified by @setIdentifier.

@Exception: A string that contains the exception message if an exception occurred during processing of the stored procedure; otherwise, it is NULL.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.45 UpdateWorkflowInstanceStatus

The UpdateWorkflowInstanceStatus stored procedure is invoked to update the WorkflowStatus attribute of a Workflow Instance .

```
PROCEDURE UpdateWorkflowInstanceStatus (
    @identifier uniqueidentifier,
    @status nvarchar(448),
    @cause uniqueidentifier,
);
```

@identifier: The Object Identifier of the Workflow Instance whose WorkflowStatus attribute is to be updated. This MUST NOT be NULL.

@status: The Workflow Instance Status value to set for the ValueString field of the WorkflowStatus attribute of the Workflow Instance. This MUST be "Created", "Running" or "Completed". This MUST NOT be NULL.

@cause: The Cause, which is the Request Object Identifier that causes this workflow status change. This MUST NOT be NULL. The server MUST assign this value to the AddCause field of the WorkflowStatus attribute of the Workflow Instance.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.46 UpdateWorkflowInstanceStatusCreated

The UpdateWorkflowInstanceStatusCreated stored procedure is invoked to set the ValueString field of the WorkflowStatus attribute of a Workflow Instance to Created.

```
PROCEDURE UpdateWorkflowInstanceStatusCreated (
@identifier uniqueidentifier,
@cause uniqueidentifier,
@hostId smallint,
@workflowDefinitionKeyWithActiveEndpoints int,
);
```

@identifier: The Object Identifier of the Workflow Instance whose WorkflowStatus is to be updated. This MUST NOT be Null.

@cause: The Cause that the server MUST assign to the AddCause field of the WorkflowStatus attribute. This MUST NOT be NULL.

@hostId: The client MUST send Host Identifier. This MUST be ignored by the server.

@workflowDefinitionKeyWithActiveEndpoints: The client MUST send NULL. This MUST be ignored by the server.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.5.47 UpdateWorkflowInstanceStatusFinal

The UpdateWorkflowInstanceStatusFinal stored procedure is invoked to set the WorkflowStatus attribute of the Workflow Instance when the workflow execution has been completed.

```
PROCEDURE UpdateWorkflowInstanceStatusFinal (
@identifier uniqueidentifier,
@status tinyint,
@cause uniqueidentifier,
@exception nvarchar(max) = null,
@workflowTypeName nvarchar(448) OUTPUT,
@requestIdentifier uniqueidentifier OUTPUT,
@requestStatusMessage nvarchar(448) OUTPUT,
);
```

@identifier: The Object Identifier of the Workflow Instance whose WorkflowStatus is to be updated. This MUST NOT be NULL.

@status: The Workflow Instance Status key. The server MUST convert the Workflow Instance Status Key to the corresponding Workflow Instance Status, as defined by being in the same row in table Workflow Instance Status Key (section 2.2.1.16). The server MUST set the ValueString field of the WorkflowStatus attribute of the WorkflowInstance that is identified by @identifier to the result of this conversion.

@cause: The Cause. The server MUST set the AddCause field of the WorkflowStatus attribute of this Workflow Instance to this value. This MUST NOT be NULL.

@exception: If an Exception occurred on the client while processing the workflow, this contains the Exception message details; otherwise, it is NULL. If the value is not NULL, the server MUST add a WorkflowStatusDetail attribute to the Workflow Instance and assign @exception to the ValueText field of that attribute.

@workflowTypeName: The type of workflow instance. The server MUST return Actionworkflowinstance.

@requestIdentifier: The Object Identifier of the Request that created the Workflow Instance identified by @identifier. The server MUST assign the ValueReference of the Request attribute to @requestIdentifier.

@requestStatusMessage: The status of the Request. The server MUST set this to the ValueString of the RequestStatus attribute of the Request that is identified by @requestIdentifier.

Return Values: An integer that MUST be 0.

Result Sets: This MUST NOT return any result sets.

3.1.6 Timer Events

If the SQL Server timeout event is triggered, the stored procedure is terminated and the call fails.

3.1.7 Other Local Events

3.1.7.1 MatchRequestParameterToManagementPolicyRule

The MatchRequestParameterToManagementPolicyRule local event returns a list of the Object Identifiers for all Management Policy Rules that match the given RequestParameter.

The input parameters are RequestParameter, RequestOperation, and Creator.

A RequestParameter matches a Management Policy Rule if:

- The RequestParameter and RequestOperation match the ActionType of the Management Policy Rule, and
- The RequestParameter matches the ActionParameters of the Management Policy Rule, and
- The Creator matches the PrincipalSet of the Management Policy Rule, and
- The valueBoolean field of the Disabled attribute of the Management Policy Rule is 1.

A RequestParameter and RequestOperation matches the ActionType of a ManagementPolicyRule if the combination of RequestOperation, RequestParameter Type, and RequestParameter Mode map to a Management Policy Rule ActionType value, as defined by the following table, and the ActionType of the ManagementPolicyRule contains the value of that mapping.

Request RequestOperation	RequestParameter Type	RequestParameter Mode	Management Policy Rule ActionType
Create	CreateRequestParameter	Not Present	Create
Create	CreateRequestParameter	Insert	Add
Put	CreateRequestParameter	Insert	Add
Put	CreateRequestParameter	Not Present	Create
Put	UpdateRequestParameter	Modify	Modify
Put	UpdateRequestParameter	Remove	Remove
Delete	N/A	N/A	Delete, Remove
Get	N/A	N/A	Read

A RequestParameter matches the ActionParameter of a Management Policy Rule if RequestParameter.PropertyName value equals the value of one of the elements of the ValueList that is contained in the ValueString field of the ActionParameter attribute or, if the ValueString field is <v>*</v>.

A Creator matches the PrincipalSet attribute of a Management Policy Rule if the Set identified by the valueReference field of the PrincipalSet attribute of the Management Policy Rule contains a ComputedMember attribute whose valueReference field equals the valueReference field of the Creator attribute of the Request.

3.1.7.2 PermissionEvaluation

This local event determines whether the given Request has permission to be processed.

The input parameter is the Request.

This returns 1 if every RequestParameter of the given Request matches at least one Management Policy Rule, as described in section 3.1.7.1. MatchRequestParameterToManagementPolicyRule (section 3.1.7.1) and the matching Management Policy Rule have 1 in the ValueBoolean field of the GrantRight attribute.

3.1.7.3 Set Transition Evaluation

This local event determines which Sets, if any, that the Target object is joining or leaving.

The input to this event is the Target Object Identifier and the RequestParameters of a Request.

It returns an array that contains the Object Identifiers of the Sets that the target object is joining and an array that contains the Sets the target object is leaving.

A Resource is joining a Set if one of the attributes added to the Resource matches the Set.

A Resource is leaving a Set if one of the attributes deleted from the Resource matches the Set.

An attribute matches a Set if it matches any Membership Condition of the Set.

An attribute matches a Membership Condition if the attribute matches all of the Membership Condition Statements of the Membership Condition.

An attribute matches a Membership Condition Statement if the Attribute Type Name of the attribute equals the Attribute field of the Membership Condition Statement and the valueString field of the attribute equals the valueLiteralString of the Membership Condition Statement.

3.1.7.4 Set Transition

This local event updates the ComputedMembers attribute of each Set that the Target Object of a Request is joining or leaving.

The input to this event is a Request Object Identifier, Cause, and Time.

Set Transition calls the Set Transition Evaluation local event.

For each Set that the Target Object of the Request is joining, the server MUST:

- Create a ComputedMember attribute where the valueReference field is set to the Target Object Identifier, the AddCause field is set to Cause, AddTime field is set to Time, and the ObjectId field is set to the Object Identifier of the Set.
- Create and return a SetTransition element where the ResourceIdentifier element value is the value of the ValueReference field of the Target attribute of the Request, the SetIdentifier element value is the Object Identifier of the Set that the Target Object is joining, the Join element value is 1, the Leave element value is 0, and the ActionWorkflowDefinition element value is set to NULL.
- Create a SetTransition attribute for the Request identified by the Request Object Identifier, where the ValueText field is set to the SetTransition element just created, the AddCause field is set to Cause, and the AddTime field is set to Time.

For each Set that the Target Object is leaving, the server MUST:

- Locate the ComputedMember attribute whose valueReference field equals the Target Object Identifier, set the deleteCause to Cause, and set the deleteTime to Time.

- Create and return a SetTransition element where the ResourceIdentifier element value is the value of the ValueReference field of the Target attribute of the Request, the SetIdentifier element value is the Object Identifier of the Set that the Target Object is joining, the Join element value is 0, the Leave element value is 1, and the ActionWorkflowDefinition element value is set to NULL.
- Create a SetTransition attribute for the Request that is identified by the Request Object Identifier, where the ValueText field is set to the newly created SetTransition element, the AddCause field is set to Cause, and the AddTime field is set to Time.

3.1.7.5 Workflow Instance Creation

The Workflow Instance Creation local event creates Workflow Instances for the Request.

The input parameter is a Request Parameter.

For each Management Policy Rule that matches at least one Request Parameter, as described in section 3.1.7.1, if the ValueReference field of the ActionWorkflowDefinition attribute is not NULL, the server MUST create a Workflow Instance where the valueReference field of the WorkflowDefinition attribute is set to the valueReference field of the ActionWorkflowDefinition.

3.1.7.6 Business Rule Evaluation

The Business Rule Evaluation verifies the following conditions to be valid; otherwise, it raises an exception:

1. The Person AccountName attribute and Domain attribute Combination MUST be unique within the ObjectsIntenal table. The Target Object single valued attribute MUST have single values only.
2. If executing a Request would result in the Target Object of the Request being in violation of the schema. A Target Object would be in violation of the schema if:
    1. The Target Object has more than one instance of a single-valued attribute, as defined by a valueBoolean field of 0 for the Multivalued attribute of the Attribute Type Description.
    2. The Target Object is missing a required attribute, as defined by the BindingDescription for the Resource type of the Target Object.

3.2 Client Details

The User Profile Synchronization Synchronization Service acts as a client when it calls the User Profile Synchronization Configuration Database that requests the execution of the stored procedures.

3.2.1 Abstract Data Model

The client maintains the following states:

client host identifier: An integer that contains identification to the client that is assigned through a call to the RegisterServer stored procedure.

maximum Requests to be processed Per Minute: An integer value that represents the maximum number of Workflow Instances that the client is willing to process in one minute.

ping Interval: An integer that specifies the expected interval, in seconds, between client calls to the RetreiveWorkflowDataForHostActivator stored procedure.

3.2.2 Timers

A connection timeout timer is set up on the client to govern the total connection time for any requests to the server. The amount of time is governed by a timeout value that is configured on the client for all server connections.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

The current protocol is triggered by the following events:

3.2.4.1 Create Person

The client calls the Create Person event to create a Person. The input parameters are the Object Identifier of the Requestor, the Request Parameters, and the host identifier.

- Create and Evaluate the Request.
    - Call EvaluateNewRequest stored procedure.
        - Using the Request Parameters as the value for the @parameters parameter.
        - Using '00000000-0000-0000-0000-000000000000' as the value for the @targetIdentifier parameter.
        - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.
        - Using 'Create' as the value for the @operator parameter.
        - Using 'Create {0}: "{1}" Request' as the value for the @displayNameTemplate.
    - Read EvaluateNewRequest result sets and parameters.
        - Read @requestIdentifier, which has the Request Object Identifier.
        - Read the EvauateNewRequest.ResultSet0, which has the ActionWorkflowInstanceIdentifier.
- Process the Request
    - Call the ProcessRequest stored procedure to process the Request.
        - Using the Request Object Identifier as the value for the @identifier parameter.
    - Read the ProcessRequest output parameter.
        - Read @resourceIdentifier, which is the Object Identifier for the created Person.
- Running Action Workflow
    - Call the UpdateRequest stored procedure to change the Request Status.
        - Using the Request Object Identifier as the value for the @identifier parameter.
        - Using 'PostProcessing' as the value for the @status parameter.

  - Using 12 as the value for the @statusKey parameter.
  - Using Request Object Identifier as the value for @cause parameter.
  - Using the client's current time as the value for @committedTime.
- Call RetrieveWorkflowInstance to retrieve the Workflow Instance and its status.
  - Using ActionWorkflowInstanceIdentifier as the value for the @identifier parameter.
  - Read RetrieveWorkflowInstance.ResultSet0, which has the DefinitionKey.
- Call RetrieveWorkflowInstanceDefinition to retrieve the Workflow Definition information of the Workflow Instance.
  - Using DefinitionKey as the value for the @workflowDefinitionKey parameter.
  - Read RetrieveWorkflowInstanceDefinition output parameter, which includes the Workflow XOML.
- Call UpdateWorkflowInstanceStatusCreated to update the Workflow Instance Status to "Created".
  - Using ActionWorkflowInstanceIdentifier as the value for the @identifier and @cause parameters.
  - Using the client host identifier for the @hostid parameter.
- Call UpdateWorkflowInstanceStatus to set the Workflow Instance Status to "Running".
  - Using ActionWorkflowInstanceIdentifier as the value for the @identifier and @cause parameters.
  - Using 0 as the value for @status.
- Construct a WorkflowInstances element where the element value is the ActionWorkflowInstanceIdentifier.
- Call UpdateRequest to update the Request with the Workflow Instance.
  - Using ActionWorkflowInstanceIdentifier as the value for @identifier.
  - Using Person Object Identifier as the value for @targetIdentifier.
  - Using WorkflowInstances element as the value for @actionWorkflowInstances.
- Call UpdateWorkflowInstanceStatusFinal to update the Workflow Instance Status to final status.
  - Using ActionWorkflowInstanceIdentifier as the value for the @identifier and @cause parameters.
  - Using 1 as the value for @status.

3.2.4.2 Add a Person to Administrators Set

The client calls this event to add a Person to the Administrators Set.

The input parameters are the Object Identifier of the Requestor, the Request Parameters, and the Object Identifier of the Set.

- Call the EvaluateNewRequest stored procedure.
  - Using the Request Parameters as the value for the @parameters parameter.
  - Using the Object Identifier of the Set as the value for the @targetIdentifier parameter.
  - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.
  - Using 'Put' as the value for the @operator parameter.
  - Using 'Update to {0}: "{1}" Request' as the value for the @displayNameTemplate.
- Read EvaluateNewRequest result sets and parameters.
  - Read @requestIdentifier, which has the Request Object Identifier.
- Call the UpdateSet stored procedure.
  - Using the Object Identifier of the Set as the value for the @setIdentifier parameter.
  - Using the Request Object Identifier as the value for the @cause parameter.
- Call CompleteRequest local event
  - Using the Request Object Identifier as the value for the Request Object Identifier parameter.

3.2.4.3 Update Management Policy Rule

The client calls this event to update the Management Policy Rule. The input parameters are the Object Identifier of the Management Policy Rule, the Object Identifier of PrincipalSet, the Object Identifier of the ResouceCurrentSet, the ObjectIdentifier of the ResourcefinalSet, the Object Identifier of the RequestParameters, and the Object Identifier of the Requestor.

- Call EvaluateNewRequest stored procedure.
  - Using the Request Parameters as value for the @parameters parameter.
  - Using the Object Identifier of the Management Policy Rule as the value for @targetIdentifier parameter.
  - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.
  - Using 'Put' as the value for the @operator parameter.
  - Using 'Update to {0}: "{1}" Request' as the value for the @displayNameTemplate.
- Read EvaluateNewRequest result sets and parameters.
  - Read @requestIdentifier which has the Request Object Identifier.
- Call UpdateManagementPolicyRule stored procedure.
  - Using the Object Identifier of the Management Policy Rule as the value for the @ObjectId parameter.

  - Using the Request Object Identifier as the value for the @cause parameter.
  - Using the Object Identifier of the PrincipalSet as the value for the @PrincipalSet parameter.
  - Using the Object Identifier of the ResourceCurrentSet as the value for the @ResourceCurrentSet parameter.
  - Using the Object Identifier of the ResourceFinalSet as the value for the @ResourceFinalSet parameter.
- Call the CompleteRequest local event.
  - Using the Request Object Identifier as the value for the Request Object Identifier parameter.

3.2.4.4 Client Register in the Server

The client calls this event to register with the server and obtain a host identifier. The input parameter is the client host name.

- Call RegisterService to register the client in the server.
  - Using client host name as the value to the @hostName parameter.
  - Read @hostId and assign to the client host identifier.

3.2.4.5 Get all Resource attributes

The client calls this event to retrieve all non-deleted attributes of a Resource. The input parameters are the Object Identifier of the Resource.

- Call GetObjectCurrent to retrieve the attributes.
  - Using the Object Identifier of the Resource as the value to the @ObjectId.
  - Read GetObjectCurrent.ResultSet0 to get the Resource Type Name from Name field.
  - Read GetObjectCurrent.ResultSet1 to get the attributes data.

3.2.4.6 Get all Resource attributes Relative to Principal

The client calls this event to retrieve, from a Resource, all non- deleted attributes that a given Principal has rights to read.

The input parameters are the Object Identifier of the Resource, the Object Identifier of the Principal, and the ValueList of the Attribute Names.

- Call CreateRequest to create a Request.
  - Using the Object Identifier of the Resource as the value to @targetIdentifier parameter.
  - Using the Object Identifier of the Principal as the value to @creator and @cause parameters.
  - Using 'Get' as the value to @operation parameter.
  - Using 'Read {0}: "{1}" Request' as the value to @displayNameTemplate parameter.
  - Read @requestIdentifier to get the Request Object Identifier.

- Call GetObjectCurrentRelative to retrieve the attributes.
  - Using the Object Identifier of the Resource as the value to @ObjectId parameter.
  - Using the Object Identifier of the Principal as the value to @Principal parameter.
  - Using the ValueList of the Attribute Names as the value to @AttributeNames parameter.
  - Read GetObjectCurrentRelative.ResultSet0 to get the Resource Type Name from the Name field.
  - Read GetObjectCurrentRelative.ResultSet1 to get the attributes data.

3.2.4.7 Modify Management Agent

The client calls this event to modify a Management Agent. This is semantically the same as Modify Management Agent event, but uses different stored procedures. The input parameters are the Object Identifier of the Requestor, the Object Identifier of the Target, the Request Parameters, and the Object Identifier for the mv-data object.

- Call EvaluateNewRequest stored procedure.
  - Using the RequestParameter as the value for the @parameters parameter.
  - Using @targetIdentifier as the value for @targetIdentifier field.
  - Using the Object Identifier of the Requestor as the value for @creator and @cause parameters.
  - Using 'Put' as the value for the @operator parameter.
  - Using 'Update to {0}: "{1}" Request' as the value for the @displayNameTemplate.
- Read EvaluateNewRequest result sets and parameters.
  - Read @requestIdentifier, which has the Request Object Identifier.
  - Read the EvauateNewRequest.ResultSet0, which has the ActionWorkflowInstanceIdentifier.
- Call EvaluateRequest.
  - Using the Request Object Identifier as the value for the @requestIdentifier parameter.
- Read EvaluateRequest result sets and parameters.
  - Read the EvauateNewRequest.ResultSet1, which has the ActionWorkflowInstanceIdentifier.
- Assign the client's current time to a dateTime variable Current Time.
- Call PreProcessUpdate
  - Using the Request Object Identifier as the value for the @identifier parameter.
  - Using the Object Identifier of the Target as the value for the @resourceIdentifier parameter.
- For each Request Parameter where the Mode is "Insert", call the stored procedure appropriate to the Request Parameter PropertyName's Attribute Data Type, as defined by the following table.

Attribute Data Type	Stored procedure
Boolean	AddObjectValueBoolean
DateTime	AddObjectValueDateTime
Integer	AddObjectValueInteger
Reference	AddObjectValueReference
String	AddObjectValueString
Text	AddObjectValueText

- Using the Object Identifier of the Target as the value for the @ObjectID parameter.
- Using the Request Parameter PropertyName as the value for the @AttributeName parameter.
- Using the Request Parameter Value as the value for the @Value parameter.
- Using the Request Object Identifier as the value for the @Cause parameter.
- Using the Current Time as the value for the @Time parameter.

- For each Request Parameter where the Mode is "Remove", call the stored procedure appropriate to the Request Parameter PropertyName's Attribute Data Type, as defined by the following table.

Attribute Data Type	Stored procedure
Boolean	DeleteObjectValueBoolean
DateTime	DeleteObjectValueDateTime
Integer	DeleteObjectValueInteger
Reference	DeleteObjectValueReference
String	DeleteObjectValueString
Text	DeleteObjectValueText

- Using the Object Identifier of the Target as the value for the @ObjectID parameter.
- Using the Request Parameter PropertyName as the value for the @AttributeName parameter.
- Using the Request Parameter Value as the value for the @Value parameter.
- Using the Request Identifier as the value for the @Cause parameter.
- Using the Current Time as the value for the @Time parameter.

- Call PostProcessUpdate.
  - Using the Object Identifier of the Target as the value for the @resourceidentifier parameter.
  - Using the Request Object Identifier as the value for the @cause parameter.
  - Using the Current Time as the value for the @Time parameter.

- Call CompleteRequest local event.
    - Using the Request Object Identifier as the value for the Request Object Identifier parameter.

3.2.4.8 Delete Management Agent

The client calls this event to delete a Management Agent. The input parameters are the Object Identifier of the Requestor, the Object Identifier of the Target, the Object Identifier of the mv-data object, and the SyncConfig-import-attribute-flow Request Parameters.

- Create and Evaluate the deletion Request.
    - Call EvaluateNewRequest stored procedure.
        - Using the Object Identifier of the Target as the value for the @targetIdentifier field.
        - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.
        - Using 'Delete' as the value for the @operator parameter.
        - Using 'Delete {0}: "{1}" Request' as the value for the @displayNameTemplate.
    - Read EvaluateNewRequest result sets and parameters.
        - Read @requestIdentifier, which has the Request Object Identifier.
- Delete the object.
    - Call DeleteObject stored procedure.
        - Using the Object Identifier of the Target as the @objectID parameter.
        - Using the Object Identifier of the Requestor as the value for the @cause parameter.
- Complete the Request.
    - Call The Complete Request local event.
        - Using the @requestIdentifier as the value for the @request identifier parameters.
- Create and Evaluate a request to modify the mv-data object.
    - Call CreateRequest.
        - Using the Object Identifier of the mv-data as the value for the @targetIdentifier parameter.
        - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.
        - Using 'Put' as the value for the @operation parameter.
        - Using 'Put {0}: "{1}" Request' as the value for the @displayNameTemplate.
        - Using the SyncConfig-import-attribute-flow Request Parameters as the value for the @parameters parameter.
    - Read CreateRequest result sets and parameters.

        - Read @requestIdentifier, which has the Request Object Identifier.
    - Call EvaluateRequest.
        - Using the Request Object Identifier as the value for the @requestIdentifier parameter.
- Complete the Request
    - Call CompleteRequest local event.
        - Using the Request Object Identifier as the value for the Request Object Identifier parameter.

3.2.4.9 Create Management Agent

The client calls this event to create a Management Agent. If there is an existing Management Agent, it is deleted. After calling this event, the client MUST call the Modify Management Agent event to complete the construction of the Management Agent. The input parameters are the Object Identifier of the Requestor, the Request Parameters, and the SyncConfig-id for the Management Agent.

- Create and Evaluate the initial Request.
    - Call EvaluateNewRequest stored procedure.
        - Using the Request Parameters as the value for the @parameters parameter.
        - Using '00000000-0000-0000-0000-000000000000' as the value for the @targetIdentifier field.
        - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.
        - Using 'Create' as the value for the @operator parameter.
        - Using 'Create {0}: "{1}" Request' as the value for the @displayNameTemplate.
    - Read EvaluateNewRequest result sets and parameters.
        - Read @requestIdentifier, which has the Request Object Identifier.
        - Read @targetIdentifier, which has the Target Object Identifier.
        - Read the EvauateNewRequest.ResultSet0, which has the ActionWorkflowInstanceIdentifier.
- Evaluate the request.
    - Call EvaluateRequest.
        - Using the Request Object Identifier as the value for the @requestIdentifier parameter.
- Delete the existing Management Agent Object.
    - Call the DeleteObject stored procedure to delete the existing Management Agent Object.
        - Using the Target Object Identifier for the @ObjectID parameter.
        - Using the Object Identifier of the Requestor as the value for the @creator and @cause parameters.

- Complete the Request.
  - Call CompleteRequest local event.
    - Using the Request Object Identifier as the value for the Request Object Identifier parameters.

3.2.5 Message Processing Events and Sequencing Rules

The client handles each stored procedure with the same processing method of calling the stored procedure synchronously and waiting for the Return Code and any returned Result Sets.

The client can execute dynamically-generated SQL queries against the stored procedures or the Tables and Views used within the User Profile Synchronization Configuration Database. However, unless otherwise specified, any addition, removal, or modification of data MUST occur only by calling the listed stored procedures. SQL queries MUST NOT attempt to add, remove, or update data in any Table or View in the User Profile Synchronization Configuration Database, unless explicitly described in this section.

3.2.6 Timer Events

3.2.6.1 SQL Connection Timeout

If the SQL Client connection timeout event is triggered, the connection and the stored procedure call fails.

3.2.6.2 Update Request Status for Completed Action Workflows Timer

The client starts a timer to periodically find Workflow Instances that have completed and to update the Request Status of their associated Requests.

When the timer event is triggered, the client does the following:

- Call UpdateRequestStatusForCompletedActionWorkflows.
  - Using maximum Requests to be processed Per Minute as the value to the @maxRequestsToProcess parameter.

3.2.6.3 Retrieve Workflow Data For Host Activator Timer

The client starts a timer to periodically retrieve Workflow Instances that are required to be run.

When the timer event is triggered, the client does the following:

- Call UpdateRequestStatusForCompletedActionWorkflows.
  - Using the client host identifier as the value to the @hostId parameter.
  - Using the ping interval as the value to the @pingIntervalSecs parameter.
  - Using '0' as the value to the @activeHostedWorkflowDefinitionSequenceNumber parameter.
  - Using '60' as the value for the @workflowControlMessagesMaxPerMinute parameters.
  - Using '30' as the value for the @RequestRecoveryMaxPerMinute parameter.
  - Using '120' as the value for the @RequestCleanupMaxPerMinute parameter.

  - Using '0' as the value for the @runRequestRecoveryScan parameter.

3.2.7 Other Local Events

3.2.7.1 CompleteRequest

The client calls this local event to complete a request. The input parameter is the Request Identifier.

- Call UpdateRequest to set the Request Status to Committed.
  - Using the Request Identifier as the value for the @identifier and @cause parameters.
  - Using "Committed" as the value for the @status parameter.
  - Ignore the result sets.
- Call UpdateRequest to set the Request Status to PostProcessing.
  - Using the Request Identifier as the value for the @identifier and @cause parameters.
  - "Using PostProcessing" as the value for the @status parameter.
  - Ignore the result sets.
- Call UpdateRequest to set the Request Status to Completed.
  - Using the Request Identifier as the value for the @identifier and @cause parameters.
  - Using "Completed" as the value for the @status parameter.
  - Ignore the result sets.

4 Protocol Examples

This section provides specific example scenarios for manipulating Administrators and Sync Engine Configuration data. These examples describe in detail the process of communication between the front-end web server (User Profile Synchronization Configuration Service) and the back-end database server (User Profile Synchronization Configuration Database).

4.1 Administrators Examples

4.1.1 Create Person

This example describes the request made to create a new Person. Because this action causes a create user request, a side effect is that another request is created to modify the domain configuration object reference to that user, as shown in the following diagram.

The client processes create user and initiates the following steps.

1. The Client calls EvaluateNewRequest to create a new request to create the Person and evaluate it.

2. EvaluateNewRequest records the request in the request table and evaluates it. The evaluation includes authenticating that the caller has rights for this operation. It returns an identifier for the created request and result sets, which have the Workflow Definition of updating the person domain configuration.

3. The Client calls ProcessRequest to process the request by passing the request identifier that was returned from the previous step.

4. ProcessRequest persists the data and returns the new object identifier of the newly created person.

5. The Client calls UpdateRequest to set the request status to be "PostProcessing".

6. The UpdateRequest sets the status and returns the result sets, all of which are empty.

7. The Client calls RetrieveWorkflowInstance to retrieve the work instance and its status that has been identified by the previous step.

8. RetrieveWorkflowInstance returns the workflow instance information.

9. The Client calls RetrieveWorkflowInstanceDefinition to retrieve the workflow instance definition information for that workflow instance identifier that is loaded by the previous step.

10. RetrieveWorkflowInstanceDefinition returns the Workflow Definition.

11. The Client calls UpdateWorkflowInstanceStatusCreated to set the status to "Created".

12. UpdateWorkflowInstanceStatusCreated persists the status.

13. The Client calls UpdateWorkflowInstanceStatus to set the workflow instance to "Running".

14. UpdateWorkflowInstanceStatus persists the status.

15. The Client calls UpdateRequest to update the request with the current workflow instance that running by passing the workflow instance identifier.

16. The UpdateRequest updates the request.

17. The Client calls the UpdateWorkflowInstanceStatusFinal to update the workflow instance to final status.

18. The UpdateWorkflowInstanceStatusFinal sets the workflow instance status.

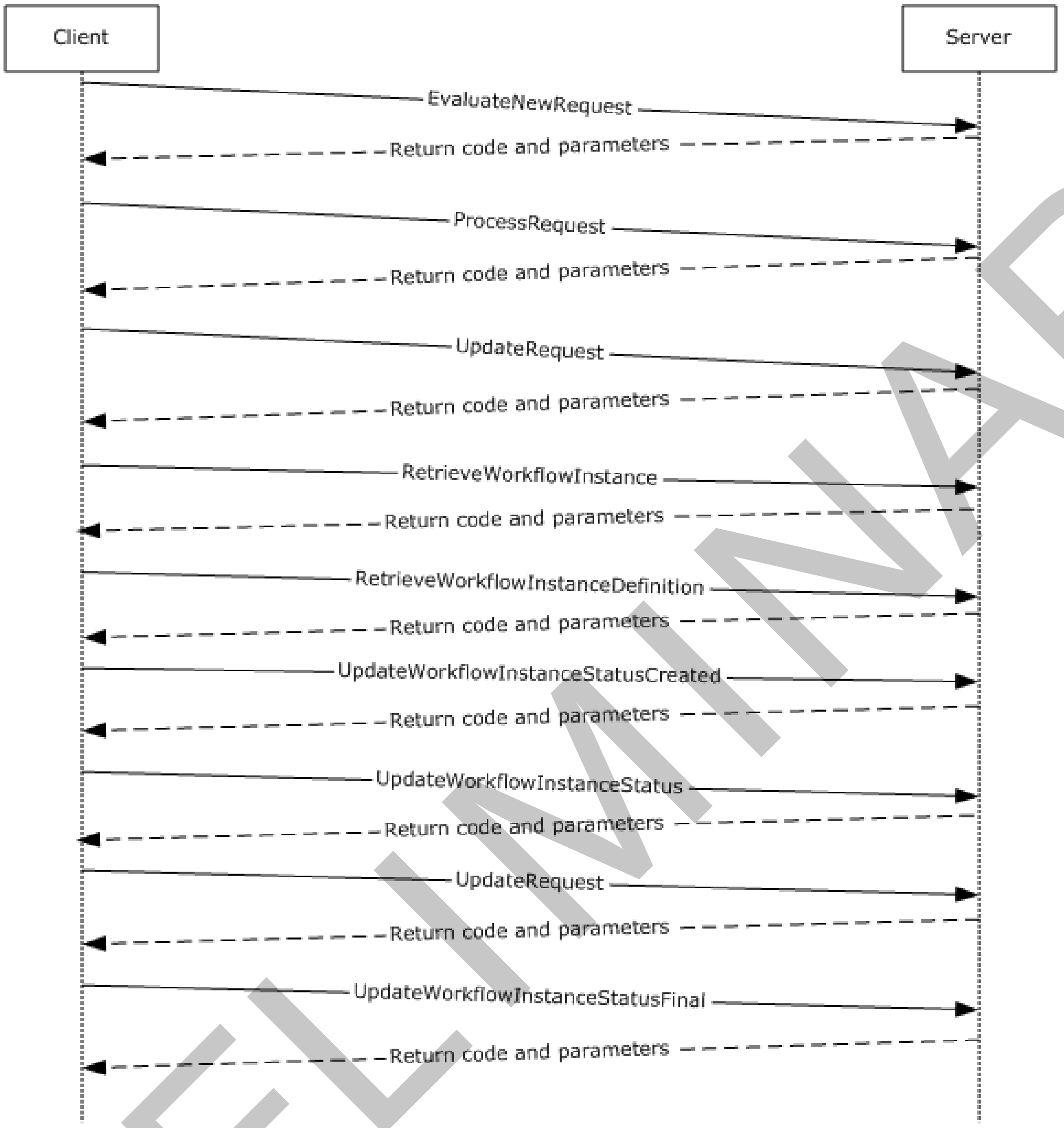
Client
Server
EvaluateNewRequest
Return code and parameters
ProcessRequest
Return code and parameters
UpdateRequest
Return code and parameters
RetrieveWorkflowInstance
Return code and parameters
RetrieveWorkflowInstanceDefinition
Return code and parameters
UpdateWorkflowInstanceStatusCreated
Return code and parameters
UpdateWorkflowInstanceStatus
Return code and parameters
UpdateRequest
Return code and parameters
UpdateWorkflowInstanceStatusFinal
Return code and parameters

Figure 5: Create Person

4.1.2 Add Person to Sync Configuration Administrators Set

This example creates a new Sync Configuration Administrators Set. The Client makes the following calls to complete this operation:

1. The Client calls EvaluateNewRequest to create a new Request to update the Sync Configuration Set.

2. EvaluateNewRequest records the Request in the request table and evaluates it. Evaluation includes authenticating that the caller is in the administrators set. It returns an identifier for the created request and result sets, all of which are empty.

3. The Client calls UpdateSet to process the request by passing in the request id that was returned from step 1.

4. UpdateSet updates the Administrators Set based on the parameters from the request passed in step 1.

5. The Client calls UpdateRequest to set the request status to be "Committed".

6. The UpdateRequest sets the status and returns the results sets, all of which are empty.

7. The Client calls UpdateRequest to set the request status to be "PostProcessing".

8. The UpdateRequest sets the status and returns the result sets, all of which are empty.

9. The Client calls UpdateRequest to set the request status to be "Completed".

10.The UpdateRequest sets the status and returns the result sets, all of which are empty.

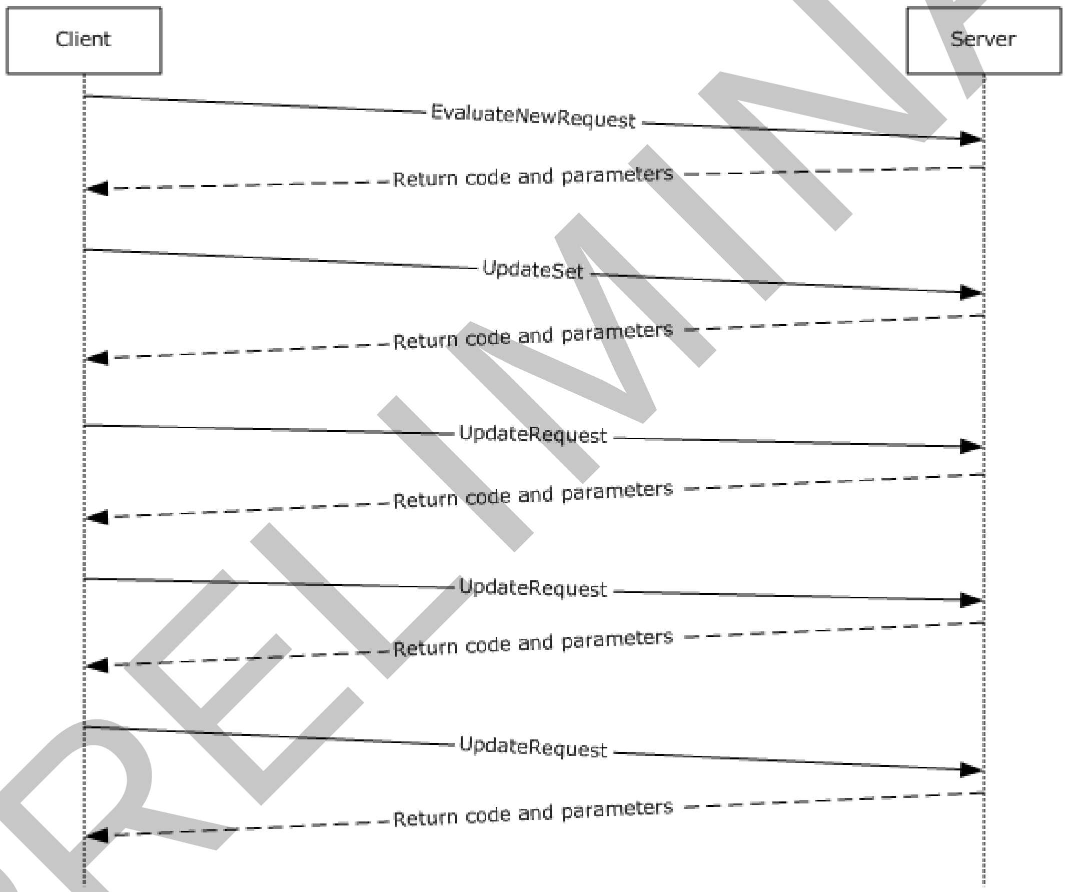
Client
Server
EvaluateNewRequest
Return code and parameters
UpdateSet
Return code and parameters
UpdateRequest
Return code and parameters
UpdateRequest
Return code and parameters
UpdateRequest
Return code and parameters

Figure 6: Add Person to Sync Configuration Administrators Set

4.2 Configurations Examples

4.2.1 Read Metaverse Configuration Example

The Read Metaverse Configuration example demonstrates reading a metaverse configuration object from the database. The client makes the following calls to complete this operation:

1. The Client calls CreateRequest to create a new request for reading an object and sending the object id.
2. CreateRequest records the request in the request table. It returns an identifier for the created request and result sets, all of which are empty.
3. The Client calls GetObjectCurrentRelative to retrieve the metaverse object and sending the metaverse object identifier and the principal user object identifier as parameters.
4. GetObjectCurrentRelative returns the result sets of the object data that are permissible to that user identifier.

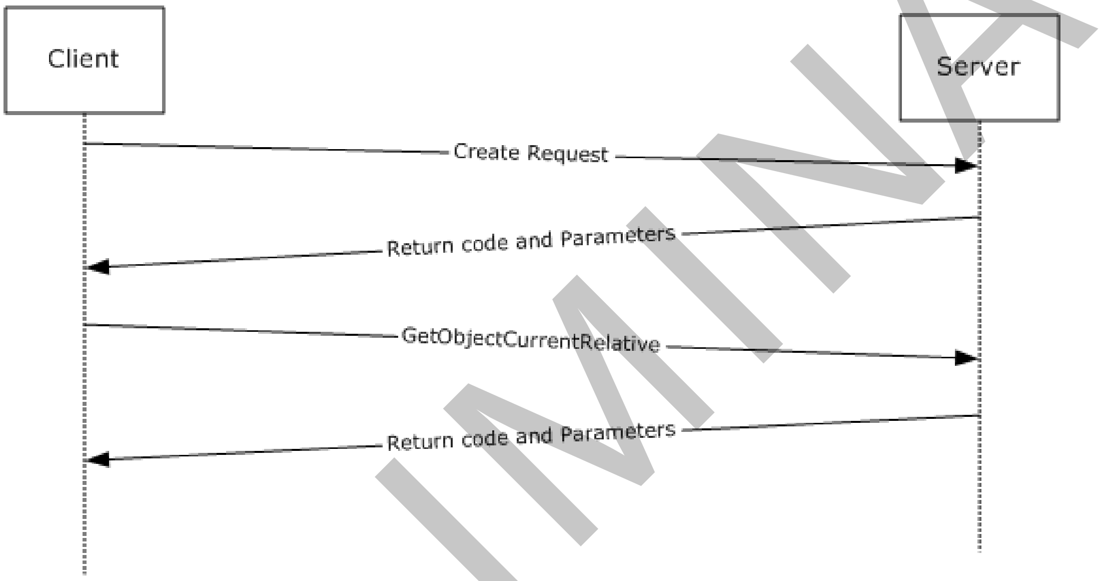

Figure 7: Read Metaverse Configuration

4.2.2 Create Management Agent Data Example

The following example demonstrates the creation of management agent data (ma-data). The client makes the following calls to complete this operation:

1. The Client calls EvaluateNewRequest to create a new request to create a Management Agent Data.
2. The EvaluateNewRequest records the request in the request table and evaluates it. Evaluation includes authenticating that the caller has rights for this operation. It returns an identifier for the created request and result sets, all of which are empty.
3. The Client calls GetObjectCurrent to retrieve this Management Agent, if it exists.

4. GetObjectCurrent returns the result sets, all of which are empty if there is no ma-data to return. Otherwise, it returns ma-data.

5. The Client calls EvaluateRequest to reevaluate the request by sending the request identifier from step 1.

6. EvaluateRequest returns the result sets, all of which are empty.

7. The Client calls UpdateRequest to recalculate the request status.

8. UpdateRequest sets the status and returns the result sets, all of which are empty.

9. The Client calls DeleteObject to delete the ma-data object, if it exists.

10. DeleteObject deletes the Object and returns.

11. The Client calls UpdateRequest to update the status to "Completed".

12. UpdateRequest sets the status and returns result sets with a child request identifier.

13. The Client calls UpdateRequest to update the status to "Committed" for the request that is identified on the previous step.

14. UpdateRequest sets the status and returns result sets, all of which are empty.

15. The Client calls UpdateRequest to update the status to "PostProcessing".

16. UpdateRequest returns code and parameters, sets the status, and returns result sets, all of which are empty.

17. The Client calls UpdateRequest to update the status to "Completed".

18. UpdateRequest sets the status and returns result sets, all of which are empty.

19. The Client calls EvaluateNewRequest to create a new request for updating the previous created ma-data from the current values that are stored on the synch engine.

20. The EvaluateNewRequest records the request and evaluates it.

21. The Client calls GetObjectCurrent to retrieve the current ma-data state.

22. GetObjectCurrent returns the ma-data.

23. The Client calls CreateRequest to create a request to update more values to the current ma-data by taking the new values as parameters.

24. The CreateRequest returns the new request identifier.

25. The Client calls EvaluateRequest to evaluate the new request that was created from the previous step.

26. The EvaluateRequest evaluates the request and returns the result sets, all of which are empty.

27. The Client calls PreProcessUpdate to validate that the newly created and updated values do not prohibit any business rule.

28. The PreProcessUpdate returns code 0.

29. The Client calls DeleteObjectValue* to delete the old values on the modified attributes.

30.DeleteObjectValue* deletes the attribute values and returns code 0.

31.The Client calls AddObjectValue* to set the attributes with the new values.

32.The AddObjectValue* returns code 0.

33.The Client calls PostProcessUpdate to apply any Set Transitions that occur as a result of the persisted data.

34.The PostProcessUpdate returns code 0.

35.The Client calls UpdateRequest to set the request status to "Completed".

36.The UpdateRequest sets the status and returns the result sets, all of which are empty.

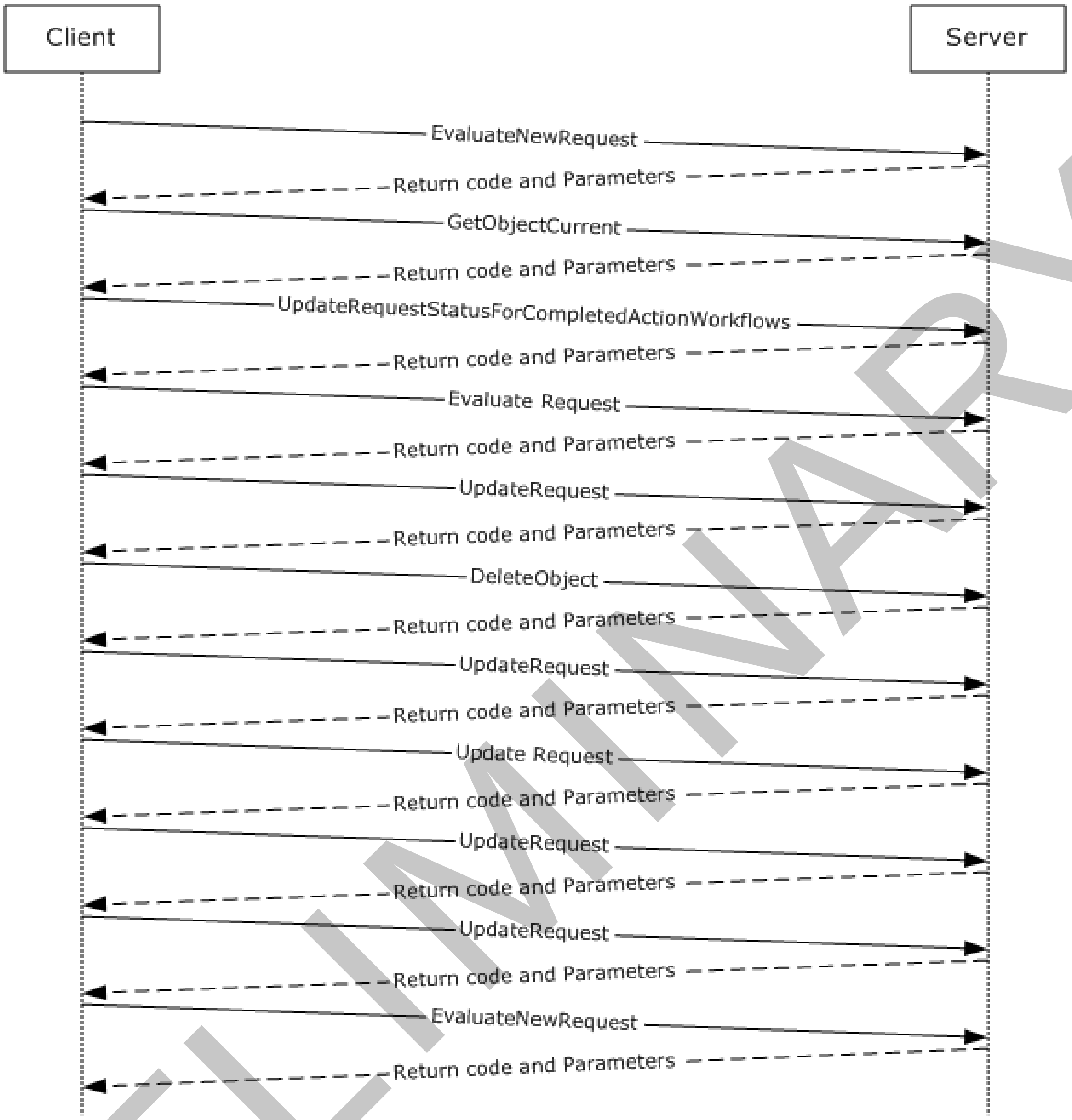
Client
Server
EvaluateNewRequest
Return code and Parameters
GetObjectCurrent
Return code and Parameters
UpdateRequestStatusForCompletedActionWorkflows
Return code and Parameters
Evaluate Request
Return code and Parameters
UpdateRequest
Return code and Parameters
DeleteObject
Return code and Parameters
UpdateRequest
Return code and Parameters
Update Request
Return code and Parameters
UpdateRequest
Return code and Parameters
UpdateRequest
Return code and Parameters
EvaluateNewRequest
Return code and Parameters

Figure 8: Create Management Agent Data

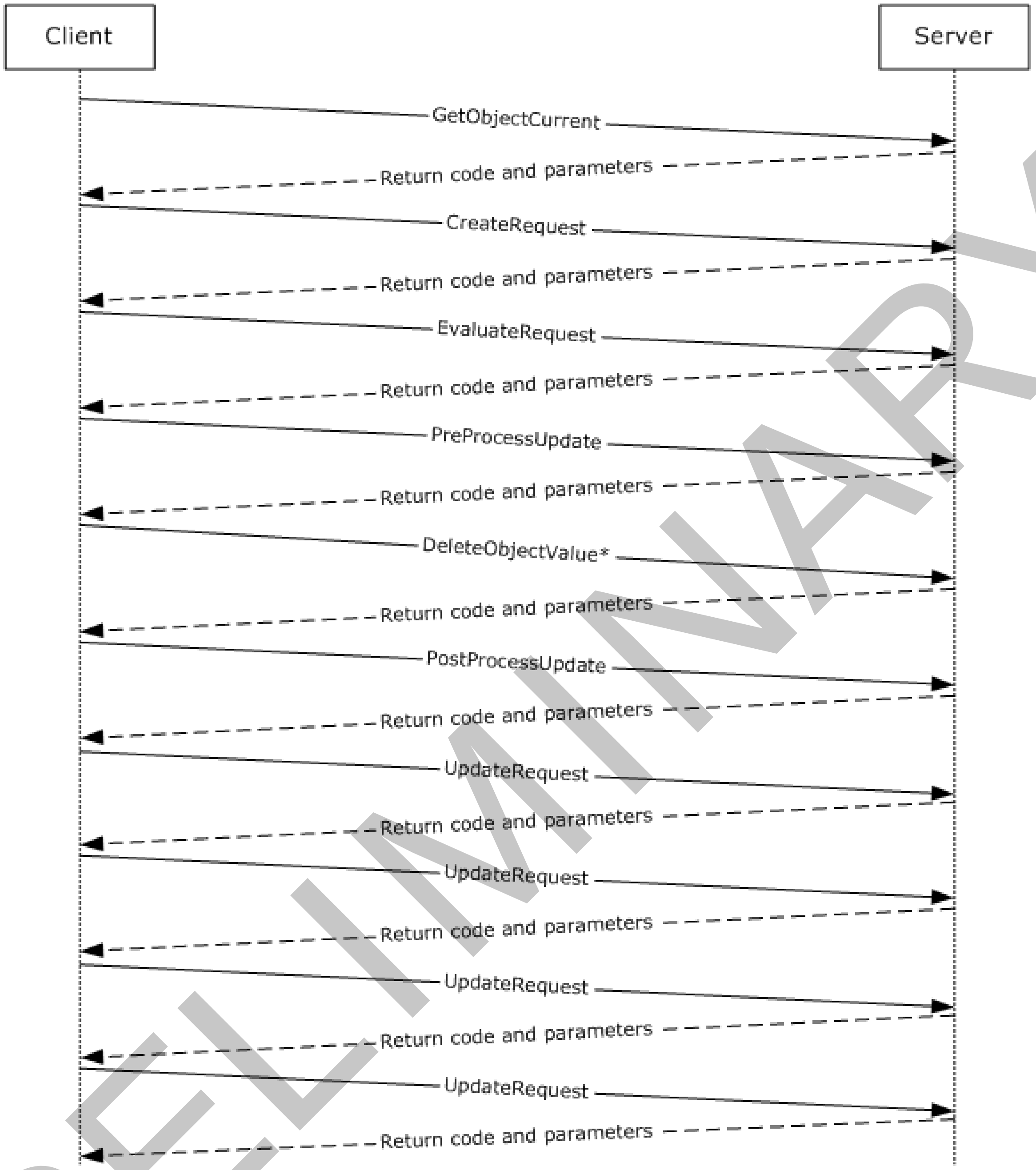
Client
Server
GetObjectCurrent
Return code and parameters
CreateRequest
Return code and parameters
EvaluateRequest
Return code and parameters
PreProcessUpdate
Return code and parameters
DeleteObjectValue*
Return code and parameters
PostProcessUpdate
Return code and parameters
UpdateRequest
Return code and parameters
UpdateRequest
Return code and parameters
UpdateRequest
Return code and parameters
UpdateRequest
Return code and parameters

Figure 9: Create Management Agent Data

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 calling the stored procedure.

5.2 Index of Security Parameters

None.

6 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® 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.

7 Change Tracking

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

8 Index