[MS-CAPR]:

Central Access Policy Identifier (ID) Retrieval Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft Open Specification Promise or the Community Promise. If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks**. The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
12/16/2011	1.0	New	Released new document.
3/30/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	2.0	Major	Significantly changed the technical content.
1/31/2013	2.0	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	3.0	Major	Significantly changed the technical content.
11/14/2013	3.0	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	3.0	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	3.0	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	4.0	Major	Significantly changed the technical content.

Table of Contents

1 Introduction

The Central Access Policy Identifier (ID) Retrieval Protocol enables an administrative tool to query the Central Access Policies (CAPs) configured on a remote computer.

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 [RFC2119]. Sections 1.5 and 1.9 are also normative but do not contain those terms. All other sections and examples in this specification are informative.

1.1 Glossary

The following terms are specific to this document:

remote procedure call (RPC): A context-dependent term commonly overloaded with three meanings. Note that much of the industry literature concerning RPC technologies uses this term interchangeably for any of the three meanings. Following are the three definitions: (*) The runtime environment providing remote procedure call facilities. The preferred usage for this meaning is "RPC runtime". (*) The pattern of request and response message exchange between two parties (typically, a client and a server). The preferred usage for this meaning is "RPC exchange". (*) A single message from an exchange as defined in the previous definition. The preferred usage for this term is "RPC message". For more information about RPC, see [C706].

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

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

1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

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.

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, https://www2.opengroup.org/ogsys/catalog/c706

[MS-DTYP] Microsoft Corporation, "Windows Data Types".

[MS-ERREF] Microsoft Corporation, "Windows Error Codes".

[MS-GPCAP] Microsoft Corporation, "Group Policy: Central Access Policies Protocol Extension".

[MS-LSAT] Microsoft Corporation, "Local Security Authority (Translation Methods) Remote Protocol".

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions".

[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

1.2.2 Informative References

[RFC4511] Sermersheim, J., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006, http://www.rfc-editor.org/rfc/rfc4511.txt

1.3 Overview

The Central Access Policy ID Retrieval (CAPR) Protocol is designed to allow an administrative tool running on one computer to remotely query the set of central access control policies configured on another computer.

Central access policy objects are created in Active Directory using administrative authorization tools. Selected central access policy objects are deployed to other computers using Group Policy: Central Access Policies Extension (CAPE, described in [MS-GPCAP]). Other administrative tools can then use CAPR to determine which central policy objects have been deployed to a given remote computer.

Within CAPE and CAPR, central access policies are represented by Central Access Policy IDs (CAPIDs). A CAPID is simply the SID of a central access policy object within Active Directory.

The typical use scenario is as follows. An administrative interface tool uses CAPR to obtain the CAPIDs of one or more central access policy objects. The tool then uses these CAPIDs with CAPE and Lightweight Directory Access Protocol (LDAP): The Protocol, specified in [RFC4511], to obtain detailed information about the policies. That data can then be presented to the user and manipulated in whatever manner is appropriate to the administrative interface tool, such as to perform authorization tasks.

This protocol defines one RPC call, LsarGetAvailableCAPIDs, for client applications to use. See section 3.1.4.1 for details of this call's use.

1.4 Relationship to Other Protocols

The CAPR Protocol is dependent upon **RPC** and **TCP** for its transport. CAPR is dependent on CAPE for the abstract data model elements which store CAPIDs. Although CAPR is not itself dependent on LDAP, scenarios that use CAPR typically also use LDAP to obtain central access policy data, once the relevant CAPIDs have been obtained.

No other protocol currently depends on the CAPR Protocol.

1.5 Prerequisites/Preconditions

The CAPR Protocol is an RPC interface, and as such has the prerequisites specified in [MS-RPCE] section 1.5 that are common to RPC interfaces.

The CAPR Protocol client must obtain the name of a remote computer that supports the Remote Authorization API Protocol before invoking this protocol.

1.6 Applicability Statement

This protocol is appropriate only for implementing tools to query the central access policies configured on a remote machine.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

- **Supported Transports:** This protocol uses multiple RPC Protocol Sequences, as specified in section 2.1.
- **Security and Authentication Methods:** This protocol uses the security and authentication methods specified in [MS-RPCE] section 3.2.1.4.1.
- **Capability Negotiation:** This protocol does not support negotiation of the interface version to use.

1.8 Vendor Extensible Fields

This protocol has no vendor-extensible fields and cannot be extended by any party other than Microsoft.

This protocol uses Win32 error codes as defined in [MS-ERREF] section 2.2. Vendors SHOULD reuse those values with their indicated meaning. Implementations that use any other values run the risk of having those values collide with future modifications to the Win32 error code set.

1.9 Standards Assignments

This protocol has no standards assignments.

Parameter	Value	Reference
UUID for lsacap	afc07e2e-311c-4435-808c-c483ffeec7c9	[C706]

2 Messages

2.1 Transport

This protocol uses the following RPC Protocol sequences:

- RPC over TCP/IP (See NCACN_IP_TCP in [MS-RPCE] section 2.1.1.1)
- Server Message Block (See NCACN_NP in [MS-RPCE] section 2.1.1.2)

This protocol uses the following RPC endpoints:

- Dynamic endpoints, as described in [C706] part 4.

This protocol MUST use the following interface identifier:

- lsacap interface: afc07e2e-311c-4435-808c-c483ffeec7c9

This protocol MUST use "\PIPE\lsarpc" as the RPC endpoint when using RPC over SMB.

2.2 Common Data Types

The following data types are specified in [MS-DTYP]:

Data type name	Section	Description
ULONG	2.2.50	A ULONG is a 32-bit unsigned integer (range: 0 through 4294967295 decimal). Because a ULONG is unsigned, its first bit (Most Significant Bit (MSB)) is not reserved for signing.

2.2.1 Structures

The CAPR Protocol defines the following structure:

Structure name	Section	Description
LSAPR_WRAPPED_CAPID_SET	2.2.1.1	A container for an array of LSAPR_SID_INFORMATION structures.

2.2.1.1 LSAPR_WRAPPED_CAPID_SET

The LSAPR_WRAPPED_CAPID_SET structure is a container for an array of LSAPR_SID_INFORMATION structures.

```
typedef struct  LSAPR WRAPPED CAPID SET {
    ULONG Entries;
    [size_is(Entries)] LSAPR_SID_INFORMATION* SidInfo;
} LSAPR_WRAPPED_CAPID_SET;
```

Entries: The number of elements in the SidInfo array.

SidInfo: A pointer to an array of LSAPR_SID_INFORMATION structures, as defined in [MS-LSAT] section 2.2.17.

3 Protocol Details

The Central Access Policy ID Retrieval Protocol is used to retrieve the set of central access policies that have been configured on a remote machine. This protocol is intended to be used in the implementation of administrative interfaces for viewing or managing the set of central access policies of specific resources on the remote machine.

All CAPR Protocol methods return ERROR_SUCCESS (0x00000000) on success. Otherwise, they return 32-bit nonzero Win32 error codes. For more information on Win32 error codes, see [MS-ERREF].

Unless otherwise specified, the pointer type for the CAPR protocol RPC interface is pointer_default(unique). Method calls are received at a dynamically assigned endpoint. The endpoints for the Netlogon service are negotiated by the RPC endpoint mapper. For information on dynamic endpoint assignment and endpoint negotiation, see [MS-RPCE] section 2.1.1.1.

The client side of this protocol involves no additional timers or other states and can simply pass calls directly from the higher-layer protocol or application to the transport. Similarly, results returned by the transport can be passed directly to the higher-layer protocol or application without further processing.

The CAPR protocol does not support version number negotiation. Client and server implementations of this protocol MUST be configured with a version number of 1.0. See section 6 for an example of configuring the version number.

3.1 lsacap Server Details

The following sections specify the data and state maintained by the lsacap RPC server, including details about receiving lsacap RPC methods on the server. The information in the following sections is intended to be illustrative of the protocol's specified behavior, without mandating any particular implementation. Implementations are not required to adhere to this model as long as their external behavior is consistent with the behavior specified in this document.

3.1.1 Abstract Data Model

This protocol uses the following ADM element, which is directly accessed from the Group Policy Central Access Policies Protocol Extension protocol, as described in [MS-GPCAP] section 3.1.1:

CentralAccessPoliciesList: The list of Group Policy central access policies on the remote computer.

3.1.2 Timers

None.

3.1.3 Initialization

The CAPR server implementation registers an endpoint with RPC over TCP/IP. The server MUST register the SPNEGO security support provider authentication_type constant [0x09] as the security provider used by the RPC interface, as specified in [MS-RPCE] section 3.3.3.3.1.3.

3.1.4 Message Processing Events and Sequencing Rules

This protocol defines the following RPC method.

Method	Description
LsarGetAvailableCAPIDs	Opnum: 0

3.1.4.1 LsarGetAvailableCAPIDs (Opnum 0)

This method returns a list of the CAPIDs of all the central access policies available on the specified remote machine. These identifiers are equivalent to the SIDs of the central access policy objects as they are stored in Active Directory.

```
NTSTATUS LsarGetAvailableCAPIDs(
        [in] handle_t BindingHandle,
        [out] LSAPR_WRAPPED_CAPID_SET* WrappedCAPIDs);
```

BindingHandle: A handle to an RPC binding for the specified remote machine.

WrappedCAPIDs: A pointer to LSAPR_WRAPPED_CAPID_SET, as defined in section 2.2.1.1.

Return Values:

If the method succeeds, the function MUST return 0x00000000 (ERROR_SUCCESS).

If the method fails, it MUST return a nonzero error code from the values defined in [MS-ERREF].

When processing this call, the server MUST return an LSAPR_WRAPPED_CAPID_SET constructed as follows:

1. The server MUST ensure that an authentication level identifier (as specified in [MS-RPCE] section 2.2.1.1.8) other than RPCE_C_AUTHN_LEVEL_NONE is present in the RPC message. Otherwise, the server MUST return STATUS_ACCESS_DENIED.
2. The *Entries* field of the LSAPR_WRAPPED_CAPID_SET MUST be set to the number of **CentralAccessPolicy** objects in the **CentralAccessPoliciesList** ADM element.
3. The *SidInfo* field of the LSAPR_WRAPPED_CAPID_SET structures MUST be set to an array of pointers to LSAPR_SID_INFORMATION structures. There MUST be one LSAPR_SID_INFORMATION structure in the array for each **CentralAccessPolicy** object in the **CentralAccessPoliciesList** ADM element. Each LSAPR_SID_INFORMATION structure MUST be set to the *CAPID* field of the corresponding CentralAccessPolicy object.

3.1.5 Timer Events

None.

3.1.6 Other Local Events

None.

4 Protocol Examples

None.

5 Security

5.1 Security Considerations for Implementers

Central access policies embody authorization policies used to control access to resources. Write permission on central access policies gives users the ability to modify authorization policies. Central access policies are designed to be managed centrally, not to be edited on client computers.

Where possible, implementations of this protocol should avoid storing central access policies on client computers. If an implementation is required to store central access policies on client computers, it should do so in secure locations that only system processes can access.

5.2 Index of Security Parameters

This protocol has no security parameters.

6 Appendix A: Full IDL

For ease of implementation, the full Central Access Policy ID Retrieval Protocol IDL interface is provided, where "ms-dtyp.idl" is the IDL found in [MS-DTYP] Appendix A. The syntax uses the IDL syntax extensions defined in [MS-RPCE] section 2.2.4 and 3.1.1.5.1. For example, as noted in [MS-RPCE] section 2.2.4.9, a pointer_default declaration is not required and pointer_default(unique) is assumed.

```
import "ms-dtyp.idl";
import "ms-lsat.idl";

typedef struct _LSAPR_WRAPPED_CAPID_SET {
    ULONG Entries;
    [size is(Entries)] LSAPR SID INFORMATION * SidInfo;
} LSAPR_WRAPPED_CAPID_SET;


 [uuid(afc07e2e-311c-4435-808c-c483ffeec7c9)]
[version(1.0)]
[pointer default(unique)]
[ms_union]
interface lsacap {
    NTSTATUS LsarGetAvailableCAPIDs(
        [in] handle_t BindingHandle,
        [out] LSAPR_WRAPPED_CAPID_SET * WrappedCAPIDs);
};
```

7 Appendix B: 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.

- Windows 8 operating system
- Windows Server 2012 operating system
- Windows 8.1 operating system
- Windows Server 2012 R2 operating system
- Windows 10 operating system
- Windows Server 2016 Technical Preview operating system

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.

8 Change Tracking

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

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

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

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

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

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

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

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

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

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

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

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

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
3.1.3 Initialization	72285 : Updated the value used for the authentication_type.	Y	Content update.
7 Appendix B: Product Behavior	Added Windows 10 to applicability list.	Y	Content update.

9 Index

R

S

T

V