

[MC-CCFG-Diff]:

Server Cluster: Configuration (ClusCfg) Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards as well as overviews of the interaction among each of these technologies support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **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 maycan make copies of it in order to develop implementations of the technologies that are described in the Open Specifications-~~this documentation~~ and maycan distribute portions of it in your implementations usingthat use these technologies or in your documentation as necessary to properly document the implementation. You maycan also distribute in your implementation, with or without modification, any schema, IDL'sschemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications-documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that maymight cover your implementations of the technologies described in the Open Specifications-documentation. Neither this notice nor Microsoft's delivery of thethis documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specification maySpecifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in the Open Specificationsthis documentation 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 maymight 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-mailemail addresses, logos, people, places, and events that are 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 as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does 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 documents are intended for use in conjunction with publicly available standardstandards specifications and network programming art, and assumes, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Revision Summary

Date	Revision History	Revision Class	Comments
8/10/2007	0.1	Major	Initial Availability
9/28/2007	0.2	Minor	Clarified the meaning of the technical content.
10/23/2007	0.2.1	Editorial	Changed language and formatting in the technical content.
11/30/2007	0.2.2	Editorial	Changed language and formatting in the technical content.
1/25/2008	0.2.3	Editorial	Changed language and formatting in the technical content.
3/14/2008	0.2.4	Editorial	Changed language and formatting in the technical content.
5/16/2008	0.2.5	Editorial	Changed language and formatting in the technical content.
6/20/2008	0.2.6	Editorial	Changed language and formatting in the technical content.
7/25/2008	0.2.7	Editorial	Changed language and formatting in the technical content.
8/29/2008	0.2.8	Editorial	Changed language and formatting in the technical content.
10/24/2008	0.2.9	Editorial	Changed language and formatting in the technical content.
12/5/2008	0.2.10	Editorial	Changed language and formatting in the technical content.
1/16/2009	0.2.11	Editorial	Changed language and formatting in the technical content.
2/27/2009	0.2.12	Editorial	Changed language and formatting in the technical content.
4/10/2009	0.2.13	Editorial	Changed language and formatting in the technical content.
5/22/2009	0.2.14	Editorial	Changed language and formatting in the technical content.
7/2/2009	0.2.15	Editorial	Changed language and formatting in the technical content.
8/14/2009	0.2.16	Editorial	Changed language and formatting in the technical content.
9/25/2009	0.3	Minor	Clarified the meaning of the technical content.
11/6/2009	0.3.1	Editorial	Changed language and formatting in the technical content.
12/18/2009	0.3.2	Editorial	Changed language and formatting in the technical content.
1/29/2010	1.0	Major	Updated and revised the technical content.
3/12/2010	1.0.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	1.0.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	1.0.3	Editorial	Changed language and formatting in the technical content.
7/16/2010	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
8/27/2010	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
11/19/2010	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	1.0.3	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	1.1	Minor	Clarified the meaning of the technical content.
9/23/2011	1.1	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	1.1	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2012	1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	1.1	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	1.2	Minor	Clarified the meaning of the technical content.
1/31/2013	1.2	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	1.2	None	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	1.2	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	1.2	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	1.2	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	1.2	No ChangeNone	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References	6
1.3	Overview	6
1.4	Relationship to Other Protocols	7
1.5	Prerequisites/Preconditions	7
1.6	Applicability Statement	7
1.7	Versioning and Capability Negotiation	7
1.8	Vendor-Extensible Fields	7
1.9	Standards Assignments	7
2	Messages	8
2.1	Transport	8
2.2	Common Data Types	8
3	Protocol Details	9
3.1	Server Details	9
3.1.1	Abstract Data Model	9
3.1.2	Timers	9
3.1.3	Initialization	9
3.1.4	IClusCfgAsyncEvictCleanup Server Details	9
3.1.4.1	CleanupNode (Opnum 7)	10
3.1.5	Timer Events	11
3.1.6	Other Local Events	11
3.2	Client Details	11
3.2.1	Abstract Data Model	11
3.2.2	Timers	11
3.2.3	Initialization	11
3.2.4	Message Processing Events and Sequencing Rules	11
3.2.5	Timer Events	11
3.2.6	Other Local Events	12
4	Protocol Examples	13
5	Security	14
5.1	Security Considerations for Implementers	14
5.2	Index of Security Parameters	14
6	Appendix A: Full IDL	15
7	Appendix B: Product Behavior	16
8	Change Tracking	17
9	Index	18

1 Introduction

The Server Cluster: Configuration (ClusCfg) Protocol is a **DCOM** interface, as specified in [MS-DCOM], for initiating configuration operations for a **failover cluster**. The ClusCfg Protocol allows users to restore a **node** that is no longer a configured member of a failover cluster, back to its precluster installation state.

Sections 1.5, 1.8, 1.9, 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~~This document uses the following terms ~~are specific to this document:~~

class identifier (CLSID): A GUID that identifies a software component; for instance, a DCOM object class (4) or a COM class.

cluster: A group of computers that are able to dynamically assign resource tasks among nodes in a group.

Distributed Component Object Model (DCOM): The Microsoft Component Object Model (COM) specification that defines how components communicate over networks, as specified in [MS-DCOM].

failover cluster: A set of independent computers that work together to increase the availability of services and applications. In [MS-CMRP], the term cluster is used as shorthand to mean the same thing as **failover cluster**.

Interface Definition Language (IDL): The International Standards Organization (ISO) standard language for specifying the interface for remote procedure calls. For more information, see [C706] section 4.

node: A computer system that is configured as a member of a **cluster**. That is, the computer has the necessary software installed and configured to participate in the **cluster**, and the **cluster** configuration includes this computer as a member.

opnum: An operation number or numeric identifier that is used to identify a specific **remote procedure call (RPC)** method or a method in an interface. For more information, see [C706] section 12.5.2.12 or [MS-RPCE].

registry: A local system-defined database in which applications and system components store and retrieve configuration data. It is a hierarchical data store with lightly typed elements that are logically stored in tree format. Applications use the registry API to retrieve, modify, or delete registry data. The data stored in the registry varies according to the version of Windows.

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].

universally unique identifier (UUID): A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager

entry-point vectors, and **RPC** objects. UUIDs are highly likely to be unique. UUIDs are also known as globally unique identifiers (GUIDs) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

well-known endpoint: A preassigned, network-specific, stable address for a particular client/server instance. For more information, see [C706].

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 dohelp@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-CMRP] Microsoft Corporation, "Failover Cluster: Management API (ClusAPI) Protocol".

[MS-DCOM] Microsoft Corporation, "Distributed Component Object Model (DCOM) Remote Protocol".

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

[MS-OAUT] Microsoft Corporation, "OLE Automation Protocol".

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

[MS-RRP] Microsoft Corporation, "Windows Remote Registry Protocol".

[MS-SCMR] Microsoft Corporation, "Service Control Manager Remote Protocol".

[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

[MSDN-CLUS] Microsoft Corporation, "Windows Clustering", <http://msdn.microsoft.com/en-us/library/aa373130.aspx>

1.3 Overview

The ClusCfg Protocol provides a DCOM interface that enables a client to restore a node back to its pre-cluster installation state. For more information about clustering, see [MSDN-CLUS].

1.4 Relationship to Other Protocols

This protocol relies on the Distributed Component Object Model (DCOM) Remote Protocol, as specified in [MS-DCOM], and on the Microsoft Remote Procedure Call (RPC) Extensions, as specified in [MS-RPCE].

1.5 Prerequisites/Preconditions

This protocol is implemented over DCOM and **RPC**, and as a result, has the prerequisites that are specified in the Distributed Component Object Model (DCOM) Remote Protocol [MS-DCOM] and the Remote Procedure Call Protocol Extensions [MS-RPCE] as being common to the DCOM and RPC interfaces.

It is assumed that a ClusCfg Protocol client has obtained the name or IP address of a remote computer that supports the ClusCfg Protocol before this protocol is invoked. This is done by using any implementation-specific method.

1.6 Applicability Statement

The method of the ClusCfg Protocol is specific to a Windows Server 2003 operating system failover cluster. As such, the protocol is applicable to a node that was a member of a failover cluster.

1.7 Versioning and Capability Negotiation

This document covers versioning issues in the following areas:

Supported transports: This protocol uses the DCOM technology (as specified in [MS-DCOM]), which provides capabilities to query for interface versions.

Protocol versions: This protocol is composed of the IClusCfgAsyncEvictCleanup DCOM interface version 0.0.

Security and authentication methods: As specified in [MS-DCOM] and [MS-RPCE].

Capability and versioning negotiation is handled as specified in [MS-DCOM].

1.8 Vendor-Extensible Fields

This protocol does not define any vendor-extensible fields.

This protocol uses HRESULT values as defined in [MS-ERREF] section 2.1. Vendors can define their own HRESULT values, provided they set the C bit (0x20000000) for each vendor-defined value, indicating that the value is a customer code.

1.9 Standards Assignments

No standards assignments have been received for this protocol. All values that are used in these extensions are in the private ranges that are specified in section 2.1. The following table contains the GUIDs for all the interfaces that are part of the *ClusCfg* Protocol object model.

Parameter	Value	Reference
RPC Interface UUID for IClusCfgAsyncEvictCleanup	52C80B95-C1AD-4240-8D89-72E9FA84025E	3.1.4
CLSID for ClusCfgAsyncEvictCleanup object	08F35A72-D7C4-42F4-BC81-5188E19DFA39	3.1.4

2 Messages

2.1 Transport

DCOM (as specified in [MS-DCOM]) is used as the transport protocol. The ClusCfg Protocol that is documented here relies on DCOM authentication (as specified in [MS-DCOM]) and encryption for all protocol messages. The protocol uses dynamic endpoints as allocated and managed by the DCOM infrastructure.

This protocol MUST use the UUID interface, as explained in section 1.9: 52C80B95-C1AD-4240-8D89-72E9FA84025E.

The class identifier (CLSID) of the object that implements this interface, as explained in section 1.9, is 08F35A72-D7C4-42F4-BC81-5188E19DFA39.

2.2 Common Data Types

This protocol uses the RPC base types and definitions that are specified in [C706] and [MS-RPCE], and does not define any additional data types.

3 Protocol Details

3.1 Server Details

The ClusCfg Protocol server provides a method to allow a client to restore a node that was evicted from a **cluster** to be restored to its precluster installation state. Evicting a node from a cluster is described in [MS-CMRP] section 3.1.1.6.

The following sections specify server details of the IClusCfgAsyncEvictCleanup interface of the ClusCfg Protocol including abstract data models, timers, and message processing rules.

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 explain 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 that is described in this document.

A server that implements this protocol was potentially configured as a node in a failover cluster. As such, the configuration operation **may** have left various executable and data files on the node as well as other persisted data, such as data that can be stored in a **registry**.

Configuration of a node as a member of a cluster is done by using implementation-specific methods between servers.

3.1.2 Timers

No protocol timers are required other than the internal timers that are used in RPC to implement resiliency to network outages, as specified in [MS-RPCE].

3.1.3 Initialization

The server **MUST** listen on the **well-known endpoint** that is defined for this RPC interface. For more information, see Transport (section 2.1).

3.1.4 IClusCfgAsyncEvictCleanup Server Details

This protocol **MUST** indicate to the RPC runtime that it is to perform a strict Network Data Representation (NDR) data consistency check at target level 5.0, as specified in [MS-RPCE] section 3.

The IClusCfgAsyncEvictCleanup interface is implemented by the server to allow the client to initiate the restoration of the designated server to its precluster installation state. The cleanup is accomplished through implementation-specific methods.

To receive incoming remote calls for this interface, the server **MUST** implement a DCOM object class with the CLSID ClusCfgAsyncEvictCleanup (as specified in section 1.9) by using the UUID {52C80B95-C1AD-4240-8D89-72E9FA84025} for this interface.

Methods in RPC Opnum Order

Method	Description
CleanupNode	The CleanupNode method restores the designated node to its pre-cluster installation state. Opnum: 7

Opnums 0, 1, and 2 are reserved for the QueryInterface, AddRef, and Release methods that are used by the standard COM IUnknown interface, as specified in [MS-DCOM]. Opnums 3 and 4 are not used across the network. These opnums are reserved and MUST NOT be reused by non-Microsoft implementations. Opnums 5 and 6 are reserved for the GetIDsOfNames, and Invoke methods in the IDispatch interface, as specified in [MS-OAUT].<1>

3.1.4.1 CleanupNode (Opnum 7)

The CleanupNode method removes all persistent artifacts that exist on the node after it is evicted from a cluster.

The Failover Cluster: Management API (ClusAPI) Protocol server provides a method to evict a node from a cluster. Evicting a node from a cluster is specified in [MS-CMRP], section 3.1.1.5. Once evicted, the node can be restored to its precluster installation state.

This method is idempotent. After it is invoked, the following actions MUST occur.

- The target node MUST no longer be a server for the Failover Cluster: Management API (ClusAPI) Protocol (as specified in [MS-CMRP]) until the node is reconfigured as a member of a cluster by using implementation-specific methods between servers.
- In any subsequent query of the ClusterInstallationState Registry Value, by means of the Windows Remote Registry Protocol ([MS-RRP]), as specified in [MS-CMRP] section 3.1.3.1, the server MUST return that the Value is set to 0x00000001 (eClusterInstallStateFilesCopied).
- In any subsequent calls of the Service Control Manager Remote Protocol ([MS-SCMR]) OpenService method for service name "ClusSvc", the server MUST complete with error 1060 (ERROR_SERVICE_DOES_NOT_EXIST). This behavior is in contrast to the behavior specified in the first bullet of [MS-CMRP] section 3.1.3.2.
- Reset any other implementation-specific values to their precluster installation state.

This method MUST NOT be invoked while the node is a configured member of a cluster.

```
HRESULT CleanupNode(  
    [in] BSTR bstrEvictedNodeNameIn,  
    [in] long nDelayIn,  
    [in] long nTimeoutIn  
);
```

bstrEvictedNodeNameIn: The name of the node on which cleanup is to be initiated. The client MUST provide the name of a configured node that was evicted from its cluster.

The name of the node passed in the *bstrEvictedNodeNameIn* parameter is the same as the name of the node evicted using [MS-CMRP]. How clients obtain node names is specified in the ApiOpenNode method of [MS-CMRP], section 3.1.4.67.

nDelayIn: The number of milliseconds that will elapse before cleanup is started on the target node. If some other process cleans up the target node while the delay is in progress, the delay is terminated before its expiration. If this value is zero, the node is cleaned up immediately.

nTimeoutIn: The number of milliseconds that this method will wait for cleanup to complete. This time-out is independent of the delay described previously, so if *nDelayIn* is greater than *nTimeoutIn*, this method will probably time out. However, after cleanup is initiated, cleanup will run to completion, but this method **may** not wait for it to complete.

Return Values: A signed 32-bit value that indicates return status. If the method returns a negative value, the method has failed. If the 12-bit facility code (bits 16-27) is set to 0x007, the value contains a Win32 error code (defined in [MS-ERREF]) in the lower 16 bits. A zero value or positive

values indicate success, with the lower 16 bits in positive nonzero values containing warnings or flags that are defined in the method implementation.

Return value/code	Description
0x00000000 S_OK	The call was successful.

Exceptions thrown:

No exceptions are thrown except those that are thrown by the underlying RPC protocol, as specified in [MS-RPCE].

3.1.5 Timer Events

No timer events are required except the events that are maintained in the underlying RPC transport (see section 2.1).

3.1.6 Other Local Events

No additional local events are used other than the events that are maintained in the underlying RPC transport (see section 2.1).

3.2 Client Details

The following sections specify client details of the ClusCfg Protocol, including abstract data models, timers, and message processing rules.

3.2.1 Abstract Data Model

None.

3.2.2 Timers

No protocol timers are required other than those internal ones that are used in an RPC to implement resiliency to network outages, as specified in [MS-RPCE].

3.2.3 Initialization

The client application initiates the conversation with the server by performing DCOM activation (as specified in [MS-DCOM] section 3.2.4.1.1) of the CLSID specified in section 1.9. After getting the interface pointer to the DCOM object as a result of the activation, the client application works with the object by making calls on the DCOM interface that it supports. After the conversation with the server is complete, the client application does a Release on the interface pointer.

3.2.4 Message Processing Events and Sequencing Rules

This protocol MUST indicate to the RPC runtime that it is to perform a strict Network Data Representation (NDR) data consistency check at target level 5.0, as specified in [MS-RPCE] section 3.

3.2.5 Timer Events

No protocol timer events are required on the client other than the events that are maintained in the underlying RPC transport.

3.2.6 Other Local Events

No additional local events are used on the client other than the events that are maintained in the underlying RPC transport.

4 Protocol Examples

The following example builds on the example that is provided in [MS-DCOM] section 4.1.

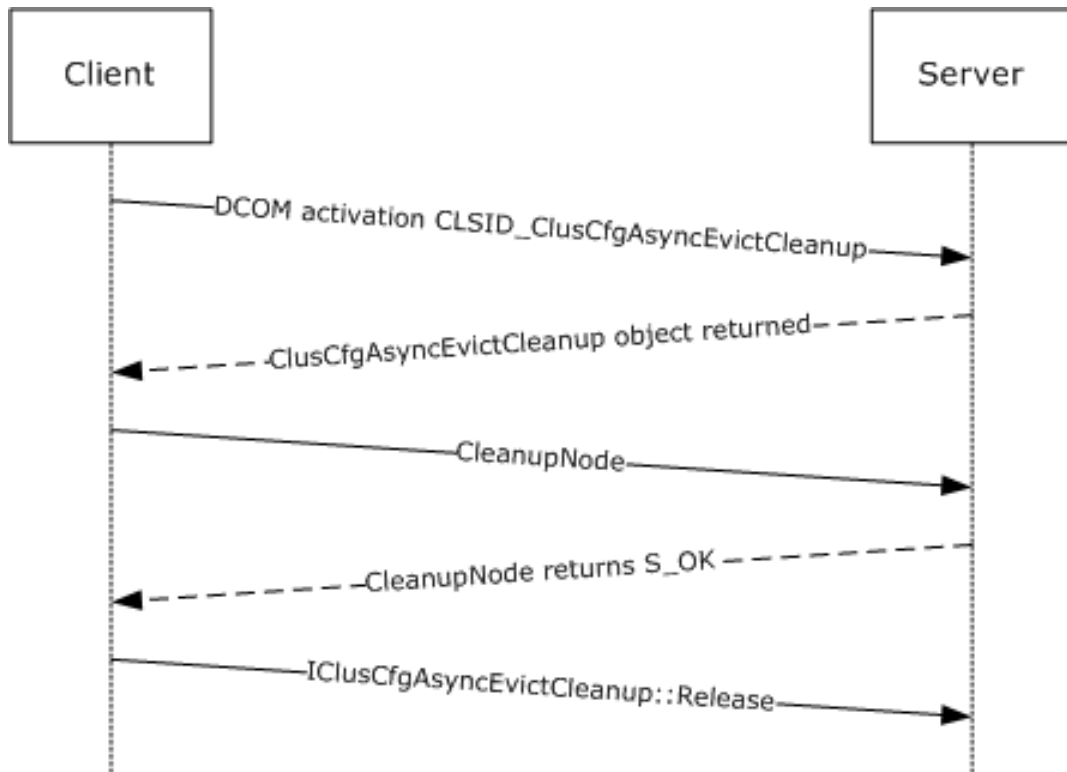


Figure 1: Cleaning up an evicted cluster node

Figure 1 shows the sequence for a client application that issues a CleanupNode call to the server.

5 Security

5.1 Security Considerations for Implementers

This protocol relies on the security features that are provided by DCOM as specified in [MS-DCOM]. ~~Implementers should review~~Review the security considerations that are listed in [MS-RPCE] section 5.1 because those considerations are also valid for DCOM and DCOM-based protocols.

5.2 Index of Security Parameters

There are no security parameters for this protocol.

6 Appendix A: Full IDL

For ease of implementation, the full **IDL** is provided where "ms-oadt.idl" is the IDL that is specified in [MS-OAUT] Appendix A.

```
import "ms-oadt.idl";

[
    object,
    uuid( 52C80B95-C1AD-4240-8D89-72E9FA84025E ),
    dual,
    helpstring("IClusCfgAsyncEvictCleanup interface"),
    pointer_default( unique )
]
interface
IClusCfgAsyncEvictCleanup : IDispatch
{
    HRESULT CleanupNode (
        [ in ] BSTR  bstrEvictedNodeNameIn,
        [ in ] long  nDelayIn,
        [ in ] long  nTimeoutIn
    );
};
```

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 Server 2003 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.

<1> Section 3.1.4: Gaps in the opnum numbering sequence apply to Windows as follows: opnums 3-4 are used only locally by Windows, never remotely.

8 Change Tracking

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

9 Index

A

- Abstract data model
 - client 11
 - server 9
- Applicability 7

C

- Capability negotiation 7
- Change tracking 17
- CleanupNode method 10
- Client
 - abstract data model 11
 - initialization 11
 - local events 12
 - message processing 11
 - overview 11
 - sequencing rules 11
 - timer events 11
 - timers 11
- Common data types 8

D

- Data model - abstract
 - client 11
 - server 9
- Data types 8
 - common - overview 8

E

- Events
 - local - client 12
 - local - server 11
 - timer - client 11
 - timer - server 11
- Examples
 - overview 13
- Examples - overview 13

F

- Fields - vendor-extensible 7
- Full IDL 15

G

- Glossary 5

I

- IClusCfgAsyncEvictCleanup interface 9
- IDL 15
- Implementer - security considerations 14
- Index of security parameters 14
- Informative references 6
- Initialization
 - client 11
 - server 9
- Introduction 5

L

Local events
 client 12
 server 11

M

Message processing
 client 11
Message processing - client 11
Messages
 common data types 8
 data types 8
 transport 8

N

Normative references 6

O

Overview (synopsis) 6

P

Parameters - security index 14
Preconditions 7
Prerequisites 7
Product behavior 16

R

References 6
 informative 6
 normative 6
Relationship to other protocols 7

S

Security
 implementer considerations 14
 parameter index 14
Sequencing rules
 client 11
Sequencing rules - client 11
Server
 abstract data model 9
 IclusCfgAsyncEvictCleanup interface 9
 initialization 9
 local events 11
 overview 9
 timer events 11
 timers 9
Standards assignments 7

T

Timer events
 client 11
 server 11
Timers
 client 11

server 9
Tracking changes 17
Transport 8

V

Vendor-extensible fields 7
Versioning 7