[MC-CCFG]:
Server Cluster:
Configuration (ClusCfg) Protocol Specification

Intellectual Property Rights Notice for Open Specifications Documentation

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

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

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

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

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

- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

Revision Summary

Date	Revision History	Revision Class	Comments
08/10/2007	0.1	Major	Initial Availability
09/28/2007	0.2	Minor	Updated the technical content.
10/23/2007	0.2.1	Editorial	Revised and edited the technical content.
11/30/2007	0.2.2	Editorial	Revised and edited the technical content.
01/25/2008	0.2.3	Editorial	Revised and edited the technical content.
03/14/2008	0.2.4	Editorial	Revised and edited the technical content.
05/16/2008	0.2.5	Editorial	Revised and edited the technical content.
06/20/2008	0.2.6	Editorial	Revised and edited the technical content.
07/25/2008	0.2.7	Editorial	Revised and edited the technical content.
08/29/2008	0.2.8	Editorial	Revised and edited the technical content.
10/24/2008	0.2.9	Editorial	Revised and edited the technical content.
12/05/2008	0.2.10	Editorial	Revised and edited the technical content.
01/16/2009	0.2.11	Editorial	Revised and edited the technical content.
02/27/2009	0.2.12	Editorial	Revised and edited the technical content.
04/10/2009	0.2.13	Editorial	Revised and edited the technical content.
05/22/2009	0.2.14	Editorial	Revised and edited the technical content.
07/02/2009	0.2.15	Editorial	Revised and edited the technical content.
08/14/2009	0.2.16	Editorial	Revised and edited the technical content.
09/25/2009	0.3	Minor	Updated the technical content.
11/06/2009	0.3.1	Editorial	Revised and edited the technical content.
12/18/2009	0.3.2	Editorial	Revised and edited the technical content.
01/29/2010	1.0	Major	Updated and revised the technical content.
03/12/2010	1.0.1	Editorial	Revised and edited the technical content.
04/23/2010	1.0.2	Editorial	Revised and edited the technical content.
06/04/2010	1.0.3	Editorial	Revised and edited the technical content.
07/16/2010	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
08/27/2010	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	1.0.3	No change	No changes to the meaning, language, or formatting of the technical content.

Contents

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 Server Cluster: Configuration (ClusCfg) Protocol allows users to restore a **node**, which is no longer a configured member of a failover cluster, back to its pre-cluster installation state.

1.1   Glossary

The following terms are defined in [MS-GLOS]:

> **active node**
> **class identifier (CLSID)**
> **cluster**
> **cluster name**
> **cluster state**
> **Distributed Component Object Model (DCOM)**
> **failover cluster**
> **node**
> **opnum**
> **registry**
> **remote procedure call (RPC)**
> **universally unique identifier (UUID)**
> **well-known endpoint**

The following terms are specific to this document:

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

1.2   References

1.2.1   Normative References

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

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, http://www.opengroup.org/public/pubs/catalog/c706.htm

[MS-CMRP] Microsoft Corporation, "Failover Cluster: Management API (ClusAPI) Protocol Specification", July 2007.

[MS-DCOM] Microsoft Corporation, "Distributed Component Object Model (DCOM) Remote Protocol Specification", March 2007.

[MS-ERREF] Microsoft Corporation, "Windows Error Codes", January 2007.

[MS-OAUT] Microsoft Corporation, "OLE Automation Protocol Specification", March 2007.

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions", January 2007.

[MS-RRP] Microsoft Corporation, "Windows Remote Registry Protocol Specification", August 2007.

[MS-SCMR] Microsoft Corporation, "Service Control Manager Remote Protocol Specification", August 2007.

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

1.2.2   Informative References

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

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

1.3   Overview

The Server Cluster: Configuration (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 Server Cluster: Configuration (ClusCfg) Protocol client has obtained the name or IP address of a remote computer that supports the Server Cluster: Configuration (ClusCfg) Protocol before this protocol is invoked. This is done by using any implementation-specific method.

1.6   Applicability Statement

The method of the Server Cluster: Configuration (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 Server Cluster: Configuration (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 Server Cluster: Configuration (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 Server Cluster: Configuration (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 Cluster: Configuration (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 Server Cluster: Configuration (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.
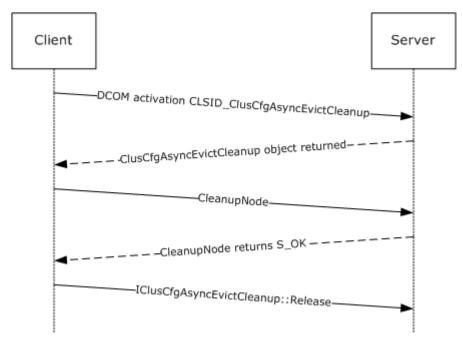
3.2.5 Timer Events

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 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-oaut.idl" is the IDL that is specified in [MS-OAUT] Appendix A.

```
import "ms-oaut.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.

Release: Friday, February 4, 2011

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